PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-093924
(43)Date of publication of application : 10.04.1998

(51)Int.Cl. H04N 5/93
G06F 3/06
G06F 3/06
G06F 13/36
G06F 17/30
G11B 20/10
H04N 7/16
H04N 7/173

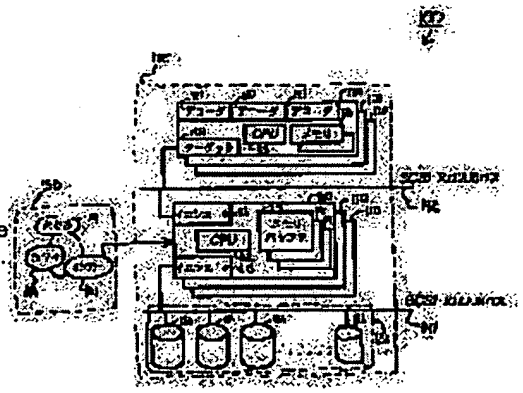
(21)Application number : 09-215258
(22)Date of filing : 08.08.1997

(71)Applicant : SHINEKKUS INF TECHNOL INC
(72)Inventor : PON-SHEN WAN
CHIN-SAN SUU

(30)Priority
Priority number : 96 692697 Priority date : 08.08.1996 Priority country : US

(54) SYSTEM AND METHOD FOR DISTRIBUTING DIGITAL DATA ON DEMAND

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an inexpensive system which can be adjusted to various conditions and which can be extended in accordance with the increase of demands by processing a command from a client, scheduling the reproduction of a video stream, managing a video file structure and controlling the flow of video data to distribution modules by respective central control modules.
SOLUTION: A video server 105 is provided with the central control modules 110, the distribution modules 120 and a storage module 130. The respective central control modules 110 receive and process a video control command from the client 101. The central control modules 110 execute a multi-task processing by using a program code stored in DRAM 232 connected to CPU 112. The program code contains a multi-processing thread executed by CPU 112 and the multi-processing thread reproduces the multiple video streams by the demand by the client 101 and execute the various control commands.



LEGAL STATUS

[Date of request for examination] 08.08.2000
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－93924

(43)公開日　平成10年(1998)４月10日

(51)Int.Cl.[6]		識別記号	FI		
Ｈ０４Ｎ	5/93		Ｈ０４Ｎ	5/93	Ｅ
Ｇ０６Ｆ	3/06	３０１	Ｇ０６Ｆ	3/06	３０１Ｅ
					３０１Ｘ
		５４０			５４０
	13/36	３１０		13/36	３１０Ｅ

審査請求　未請求　請求項の数17　ＯＬ　（全 28 頁）　最終頁に続く

(21)出願番号　特願平9－215258

(22)出願日　平成９年(1997)８月８日

(31)優先権主張番号　６９２６９７

(32)優先日　1996年８月８日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　597113332
シネックス・インフォメーション・テクノロジーズ，インコーポレイテッド
アメリカ合衆国カリフォルニア州94538，フレモント，スピナカー・コート・3797

(72)発明者　ポンーシェン・ワン
アメリカ合衆国カリフォルニア州95120，サン・ノゼ，ハンプスウッド・ウェイ・955

(72)発明者　チンーサン・スー
アメリカ合衆国カリフォルニア州95120，サン・ノゼ，エルウッド・ロード・7026

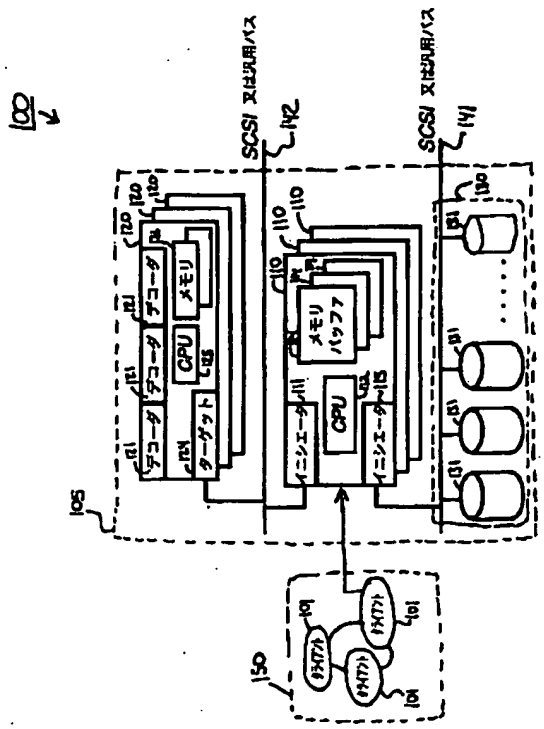
(74)代理人　弁理士　古谷　馨　（外２名）

(54)【発明の名称】　ディジタルデータをオンデマンドで分配するシステム及び方法

····【要約】

【課題】　従来の低ｺｽﾄの構成要素を用いて多数のﾋﾞﾃﾞｵｽﾄﾘｰﾑをﾘｱﾙﾀｲﾑで分配する調整可能で拡張可能なﾋﾞﾃﾞｵｻｰﾊﾞｼｽﾃﾑを提供する。

【解決手段】　複数のﾋﾞﾃﾞｵｾｸｼｮﾝを記憶する大容量記憶ﾓｼﾞｭｰﾙと、ﾀｰｹﾞｯﾄﾓｰﾄﾞで動作することで····装置をｴﾐｭﾚｰﾄする····ｲﾝﾀｰﾌｪｲｽを有する分配ﾓｼﾞｭｰﾙであって･つ又は･つ以上の処理ﾓｼﾞｭｰﾙを有しており各処理ﾓｼﾞｭｰﾙがﾋﾞﾃﾞｵｸﾗｲｱﾝﾄに分配するためのﾋﾞﾃﾞｵﾃﾞｰﾀを処理し、前記分配ﾓｼﾞｭｰﾙが前記····ｲﾝﾀｰﾌｪｲｽを介してﾋﾞﾃﾞｵﾃﾞｰﾀを受信し、及び該受信ﾋﾞﾃﾞｵﾃﾞｰﾀをﾊﾞｯﾌｧﾘﾝｸﾞし該ﾋﾞﾃﾞｵﾃﾞｰﾀをｸﾗｲｱﾝﾄに分配するようﾌﾟﾛｸﾞﾗﾑされた分配ﾓｼﾞｭｰﾙと、前記大容量記憶ﾓｼﾞｭｰﾙに接続された第･の····ｲﾝﾀｰﾌｪｲｽと、前記分配ﾓｼﾞｭｰﾙに接続された第･の····ｲﾝﾀｰﾌｪｲｽとを有する中央制御ﾓｼﾞｭｰﾙであってｸﾗｲｱﾝﾄからﾋﾞﾃﾞｵ制御ｺﾏﾝﾄﾞを受信し該受信したﾋﾞﾃﾞｵ制御ｺﾏﾝﾄﾞに応じて前記ﾋﾞﾃﾞｵｾﾚｸｼｮﾝの再生を制御する中央制御ﾓｼﾞｭｰﾙとを備えている。



【特許請求の範囲】

【請求項１】ビデオクライアントのサービスを行うためのビデオサーバシステムであって、
複数のビデオセレクションを記憶するための大容量記憶モジュールと、
ターゲットモードで動作することによりＳＣＳＩ装置をエミュレートするＳＣＳＩインターフェイスを有する分配モジュールであって、１つ又は２つ以上の処理モジュールを有しており、その各処理モジュールが、ビデオクライアントに分配するためのビデオデータを処理するためのものであり、前記分配モジュールが、前記ＳＣＳＩインターフェイスを介してビデオデータを受信し、該受信したビデオデータをバッファリングし、及び該バッファリングされたビデオデータをクライアントに分配するようにプログラムされている、分配モジュールと、
前記大容量記憶モジュールに接続された第１のＳＣＳＩインターフェイスと、前記分配モジュールに接続された第２のＳＣＳＩインターフェイスとを有する中央制御モジュールであって、クライアントからビデオ制御コマンドを受信し、該受信したビデオ制御コマンドに応じて前記ビデオセレクションの再生を制御する、中央制御モジュールとを備えていることを特徴とする、ビデオサーバシステム。

【請求項２】前記分配モジュールに含まれる少なくとも１つの処理モジュールが、圧縮されたビデオデータの圧縮解除を行うためのビデオデータ圧縮解除手段である、請求項１に記載のビデオサーバシステム。

【請求項３】前記ビデオデータ圧縮解除手段が、モーションピクチャエキスパートグループ･ＭＰＥＧ･規格に従ってデータの圧縮解除を行う、請求項２に記載のビデオサーバシステム。

【請求項４】前記分配モジュールが、ラックマウント可能なシャシにマウントされたコンピュータ用マザーボードであり、前記中央制御モジュールが、ラックマウント可能なシャシにマウントされたコンピュータ用マザーボードである、請求項１に記載のビデオサーバシステム。

【請求項５】前記分配モジュールに含まれる少なくとも１つの処理モジュールが、イーサネットネットワークを介した分配用にビデオデータをフォーマットするイーサネットフォーマットモジュールである、請求項１に記載のビデオサーバシステム。

【請求項６】前記分配モジュールに含まれる少なくとも１つの処理モジュールが、ＡＴＭネットワークを介した分配用にビデオデータをフォーマットする非同期伝送モード（ＡＴＭ）モジュールである、請求項１に記載のビデオサーバシステム。

【請求項７】１つ又は２つ以上の付加的な分配モジュールを更に備えており、
前記大容量記憶モジュールが複数のＳＣＳＩ互換の記憶装置を備えており、
前記中央制御モジュールが、前記大容量記憶装置に接続された第１の複数のＳＣＳＩインターフェイスと、１つ又は２つ以上の前記分配モジュールに各々接続された第２の複数のＳＣＳＩインターフェイスとを有している、請求項１に記載のビデオサーバシステム。

【請求項８】前記ビデオセレクションがそれぞれ制御データ及びビデオデータを含んでおり、前記制御データが、前記中央制御モジュール上で実行されるオペレーティングシステムに関連するファイル管理システムを用いて記憶され、前記ビデオデータが、前記ファイル管理システムをバイパスして生フォーマットで記憶される、請求項１に記載のビデオサーバシステム。

【請求項９】前記制御データが、記憶装置上の未使用のメモリのアドレスを示すアドレスマップと、ビデオオブジェクトの各データブロックを記憶装置上のアドレスにマッピングするアドレスマップとを含んでいる、請求項８に記載のビデオサーバシステム。

【請求項１０】前記中央制御モジュールが、前記大容量記憶装置に記憶されているビデオデータを読み出すための複数の読み出し要求を生成し、
該読み出し要求を記憶するためのメッセージ待ち行列と、
前記メッセージ待ち行列に接続されて前記読み出し要求の優先順位付けを行う優先順位付け手段とを更に備えている、請求項１に記載のビデオサーバシステム。

【請求項１１】前記優先順位付け手段が、
各読み出し要求が緊急の読み出し要求であるか否かを判定し、
緊急の各読み出し要求毎にデッドラインを計算し、
非緊急の読み出し要求をサービスすることにより緊急の読み出し要求がそのデッドラインを逸することになるか否かを判定し、
非緊急の読み出し要求をサービスすることにより緊急の読み出し要求がそのデッドラインを逸することにならないとの判定に応じて非緊急の読み出し要求をサービスし、
非緊急の読み出し要求をサービスすることにより緊急の読み出し要求がそのデッドラインを逸することになるとの判定に応じて緊急のデッドラインを有する読み出し要求をサービスする、という各ステップからなる方法を実施するものである、請求項１０に記載のビデオサーバシステム。

【請求項１２】前記大容量記憶モジュールが複数の記憶装置を備えており、
前記複数の記憶装置にわたってストライプ構成で複数の前記ビデオセレクションが記憶されており、
受信した制御コマンドに応じたビデオセレクションの再生の制御が、
複数の時間スロットを規定し、
再生が開始される記憶装置を識別し、該識別された記憶

装置に関する第１の利用可能な時間スロット内にビデオセレクションの再生をスケジューリングし、各記憶装置が複数の時間スロットのサービスをローテーションで行う、という各ステップからなる、請求項１に記載のビデオサーバシステム。

【請求項１３】前記大容量記憶モジュールが複数の記憶装置を備えており、

少なくとも１つのビデオセレクションが、複数のデータブロックを含んでおり、前記複数の記憶装置にわたってストライプ構成で記憶されており、

Ｍ個のデータブロック毎に計算されるエラーブロックを更に有しており、該Ｍが、前記ストライプ構成に使用される前記記憶装置の数よりも小さい整数の冗長ファクタである、請求項１に記載のビデオサーバシステム。

【請求項１４】時間クリティカルコマンドと非時間クリティカルコマンドとを含む複数の制御コマンドを受信するよう構成されたビデオサーバシステムにおいて、受信した制御コマンドのサービスの優先順位付けを行う方法であって、

各時間クリティカルコマンド毎にデッドラインを決定し、

時間クリティカルコマンドにそのデッドラインを逸させることなく非時間クリティカルコマンドをサービスすることができるか否かを判定し、

時間クリティカルコマンドにそのデッドラインを逸させることなく非時間クリティカルコマンドをサービスすることができるとの判定に応じて非時間クリティカルコマンドをサービスし、

時間クリティカルコマンドにそのデッドラインを逸させることなく非時間クリティカルコマンドをサービスすることができないとの判定に応じて時間クリティカルコマンドをサービスする、という各ステップからなることを特徴とする、ビデオサーバシステムにおいて受信した制御コマンドのサービスの優先順位付けを行う方法。

【請求項１５】複数（Ｎ個）の記憶装置にディジタルデータを記憶させる方法であって、

ディジタルデータをデータブロックへと分割し、

複数のデータブロックを各記憶装置に記憶させ、

Ｎ未満の整数の冗長ファクタ（Ｍ）を選択し、

Ｍ個のデータブロック毎にエラー回復ブロックを生成し、

関連するデータブロックを記憶する記憶装置とは異なる記憶装置に該エラー回復ブロックを記憶させる、という各ステップからなることを特徴とする、複数の記憶装置へのディジタルデータの記憶方法。

【請求項１６】エラー回復ブロックの生成が、パリティコードを計算することからなる、請求項１５に記載の方法。

【請求項１７】各記憶装置への複数のデータブロックの記憶が、複数の記憶装置にわたってデータブロックのストライプ処理を行うことからなる、請求項１６に記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、一般にリアルタイムサーバシステム及びプロセスに関し、特に、クライアントロケーションにビデオストリームを分配するシステム及びプロセスに関する。

【０００２】

【従来の技術】データの記憶、読み出し及び圧縮技術の改善に伴い、一般にはリアルタイムサーバシステムの使用が、特にビデオオンデマンドシステムの使用が、広く普及してきている。ビデオオンデマンド用途には、歓待施設（即ち、ホテル、モーテル、コンドミニアム、及び病院）、カラオケ（一般に音声記録の再生を伴うものであり、画像情報の再生を伴うものもある）及びインフォメーションセンターにおけるコンテンツ（即ち内容）の分配が含まれる。ビデオオンデマンドシステムは、選択されたビデオファイルを記憶し（一般に各ビデオファイルは映画、簡単な情報表示、又は他の種類のビデオコンテンツに相当する）、ユーザによる制御の下で選択されたビデオファイルを読み出す（即ち再生する）。従って、ビデオオンデマンドシステムを使用する場合、一人又は複数のユーザは、クライアントネットワークを介してビデオファイルを選択しアクセス･即ち再生･する。更に、従来のビデオオンデマンドシステムは、一般に、再生、停止、一時停止、巻戻し、及び早送り等の従来のビデオカセットレコーダ（ＶＣＲ）に見られるような様々な制御機能をユーザに提供する。ここで、「ビデオ」という用語は、音声部分及び画像部分を有するコンテンツ、又は音声のみのコンテンツ、又は画像のみのコンテンツ、又はその他の種類のディジタルコンテンツを含むものである、ということが理解されよう。

【０００３】

【発明が解決しようとする課題】ビデオオンデマンドシステムに関するチャネル要件（即ち、サーバにより分配されるビデオストリームの数）は、各々の特定のビデオオンデマンドシステムによって異なる。例えば、大きなホテルは、小さなホテルよりも多数のチャネルを必要とし、インフォメーションセンターは、そのインフォメーションセンターの場所及び提供する情報の種類に基づき、サービス対象となるクライアントの数が多かったり少なかったりする。更に、ビデオオンデマンドシステムは、始終大きなチャネル容量が要求される場所に設置される場合がある。例えば、歓待施設（即ち、ホテル、モーテル、コンドミニアム、及び病院）に設置されたシステムは、初期は少数の部屋または装置に対してサービスを行うことになるが、設備のサイズが大きくなると、または該サービスに気づいて消費者が増大すると、システムの需要が増大することになる。この問題は、おそら

く、付加的なクライアントロケーションを提供するために必要とされる物理的なインフラを回避することができないインフォメーションセンター等の用途においては更に深刻なものとなる。

【０００４】更に、ビデオオンデマンドシステムのビデオ記憶要件は、特定の用途によって異なるものとなる。例えば、歓待施設は、長編ビデオ映画を多数の選択を提供することを所望し、従って、かなり大容量の記憶要件を有することになる。一方、インフォメーションセンターは、それよりもはるかに小さな記憶要件を有する傾向にあり、情報コンテンツが長編映画と比較して小さい場合には特にそうである。

【０００５】従来の多くのビデオオンデマンドシステムは一定の高いコストアーキテクチャを有するものである。特に、従来の幾つかのビデオオンデマンドシステムは、多数のビデオストリームのリアルタイム分配を行うために、ハイエンドワークステーション又は特に高速のコンピュータを使用している。その他の従来のビデオオンデマンドシステムでは、多数のビデオストリームを分配するための処理要求に合わせるために、イベントのマルチタスク処理用の多数のプロセッサを備えたコンピュータを用いている。従来のビデオオンデマンドシステムは、ハイエンドの及び／又は特殊なハードウェアを使用するため、一般に極めて高コストのものとなる。かかる従来のビデオオンデマンドシステムは、指定された最大数のビデオストリームに適応するように一般に設計され、従って、その容量を越えて容易に拡張させることはできない、という欠点を更に有するものである。

【０００６】様々なビデオオンデマンド用途の異なる要件に合わせて調整可能であり、個々のサーバロケーションの要求の増大に合わせて拡張可能である、単一の低コストのビデオオンデマンドシステムを提供することが望ましい。

【０００７】従って、多数のビデオストリームやその他のディジタルデータストリームを並行して分配するための、調整可能及び拡張可能でコスト効率の良好な方法及びプロセスが必要とされている。

【０００８】更に、一般にコンピュータシステムの重要な部分、特にビデオオンデマンドシステムの重要な部分は、その大容量記憶装置部分である。ビデオサーバ（ビデオオンデマンド）に関して言えば、大容量記憶装置部分は、ビデオコンテンツを記憶する。別の種類のコンピュータシステムの場合、大容量記憶装置部分は、他の種類のディジタルコンテンツ、例えば、コンピュータプログラム、データベース、画像、データ等を記憶する。特定の用途がビデオオンデマンドシステムにおけるものであろうとそれ以外の種類のコンピュータシステムにおけるものであろうと、大容量記憶装置のサイズ、処理速度、及びコストは、システムの仕様、性能、及びコストに影響を及ぼすものとなる。

【０００９】従来の大容量記憶アーキテクチャには、安価なディスクドライブからなる冗長ディスクドライブアレイ（ＲＡＩＤ）を用いるものがある。従来、かかるアーキテクチャは、典型的には、従来入手可能な高性能で大きくて高コストのディスクドライブよりも小さくて安価で信頼性のあるドライブからなるドライブアレイを用いている。これらの従来のＲＡＩＤシステムの中にはストライピングを用いたものがあり、この場合には、データオブジェクトが「複数のデータストライプ」へと分割され、次いでそれらのデータストライプが、並列ディスク動作によって性能の改善を達成するためにディスクアレイ上にインターリーブされる。また、各データストライプは、ディスクアクセスを促進させるだけのサイズのデータブロックへと更に副分割される場合もある。一般に、従来のディスクアレイは、信頼性を増大させるために、ミラー又はパリティベースの機構といった形の冗長性を取り入れたものとなる。

【００１０】詳細には、従来のＲＡＩＤレベル１はミラー処理を用いており、その他の一層高レベルの従来のＲＡＩＤシステムは、エラー訂正用のパリティブロックを用いている。従来、このパリティブロックは、１つのストライプスライスにわたる（即ち、ディスクアレイにわたる）データブロックの排他ＯＲをとることによって生成される。従来、各パリティブロックは、その関連するデータストライプとは異なるディスクに記憶される。従って、ディスクの故障が生じた場合には、その故障したディスクに記憶されているデータブロックは、そのパリティブロックを使用して（そのデータストライプスライス内の他の全てのデータブロックに対応するパリティブロックの排他ＯＲ演算を行うことにより）再構成される。

【００１１】従って、Ｎ個のディスクを備えたＲＡＩＤシステムにおいて、ディスクが故障した場合には、失われた１つのデータブロックを再構成するために・Ｎ－１・個のディスクから・Ｎ－１・個のデータブロックを読み出す必要がある。・Ｎ－１・個のディスクについての読み出し動作は、サブシステムのロード性能が許容するのであれば応答時間の低減のために並列に行われるが、かかる故障発生時にはやはりかなり大きな負荷がロード性能に課されることになる。システムのディスク数・Ｎ・が増えるにつれ、故障モードにおける性能上の不利益が悪化する。従って、性能上の不利益を制限するために、ディスク数・Ｎ・を比較的小さい値に制限することが望ましい。

【００１２】一方、ＲＡＩＤシステムの高いスループットを得るために、通常のデータアクセス時には多数（Ｎ個）のディスクを有して、多数のディスク動作を並列に行うことができるようにすることが望ましい。この側面は、故障モードで少数のＮが望ましいことと矛盾する。従って、故障モードで許容不能な大きな性能上の不利益を伴うことなくシステムの信頼性及び性能を向上させる

ＲＡＩＤシステム及び方法が必要とされている。

【００１３】

【課題を解決するための手段】本発明によれば、従来の低コストの構成要素を用いて多数のビデオストリームをリアルタイムで分配する、調整可能で拡張可能なビデオサーバシステムが提供される。このシステムは、１つ又は２つ以上の中央制御モジュール（ＣＣＭ）と、１つ又は２つ以上の分配モジュール（ＤＭ）と、１つ又は２つ以上の記憶モジュール（ＳＭ）とを備えている。各中央制御モジュールは、従来の２つのスモールコンピュータシリアルインターフェイス（ＳＣＳＩ）コントローラカードを備えた従来のコンピュータであり、その各ＳＣＳＩコントローラカードは、「開始……」モードで動作して、１つ又は２つ以上の分配モジュール及び記憶モジュールとのインターフェイスをとる。各中央制御モジュールはまたローカルメモリを備えており、このローカルメモリは、分配モジュールへの分配に先立って記憶モジュールから取り出されたデータを記憶するための中間バッファメモリとして使用される。各中央制御モジュールは更に、１つのユーザ（クライアント）または１つのクライアントネットワークとの結合を行うための通信インターフェイスを備えている。各中央制御モジュールは、クライアントから受信したコマンドを処理し、多数のビデオストリームの再生に関するスケジューリングを行い、ビデオファイル構造を管理し、１つ又は複数の分配モジュールへのビデオデータの流れを制御して、リアルタイム再生を確実にする。

【００１４】また、各分配モジュールは、「ターゲット」モードで動作する従来のＳＣＳＩコントローラカードを備えた従来のコンピュータである。ＳＣＳＩコントローラカードを備えていることに加え、各分配モジュールは、クライアントへの分配に先立ってビデオストリームの処理を行う１つ又は２つ以上の処理モジュールを備えている。一実施例では、該処理モジュールはビデオデコーダであり、その各々は、ビデオデータストリームの解凍のためのものである。この実施例の場合、該ビデオデコーダは、従来のＭＰＥＧ－１又はＭＰＥＧ－２デコーダである。

【００１５】別の実施例では、該処理モジュールは、従来のネットワークインターフェイスカードであり、該カードは、ビデオデータストリームのフォーマッティングや、イーサネット、ＡＴＭ、又はＰＳＴＮネットワーク等を介したクライアントへのビデオデータストリームの供給を行うものである。更に、各分配モジュールは、分配モジュール上での処理に先立ってビデオデータを記憶するためのビデオバッファとして使用されるローカルメモリを備えている。

【００１６】各記憶モジュールは大容量記憶媒体であり、該大容量記憶媒体は、ビデオデータ等のディジタル情報を記憶するよう構成されたものであり、標準的なＳＣＳＩプロトコルを用いて中央制御モジュールによってアクセスされる。各記憶モジュールは、例えば、ハードディスク、又はＣＤ･ＲＯＭドライブ、又はハードディスク列……、又はＣＤ･ＲＯＭ列、又はその他の種類の大容量記憶媒体である。

【００１７】更に、本発明によれば、中央制御モジュールは、ハイブリッドファイル管理スキームを用いて、データアクセス性能の向上及びメモリ利用の改善を達成する。該ハイブリッドファイル管理スキームは、中央制御モジュール上で動作する従来のオペレーティングシステムと、記憶装置に記憶されている未処理ビデオデータの制御及びアクセスを直接行うための、従来のファイルマネージャをバイパスするカスタマイズされたファイル管理ソフトウェアとの両方を用いるものである。このハイブリッドスキームは、未処理のビデオデータ並びにビデオ記憶マップに関する制御情報を管理するためにオペレーティングシステムのファイル管理サービスを利用するビデオデータに対するアクセスタイムを最適化する。

【００１８】本発明の別の側面によれば、中央制御モジュールは、優先順位法…… ……を実施して、サーバシステムにより生成される複数のビデオデータストリームの間で、各記憶モジュールに含まれる記憶装置のアクセスの優先順位付けを行う。複数のビデオデータストリームによって複数の読み出し要求が生成される度に、優先順位法は、その読み出し要求（読み出しメッセージ）が緊急のものか非緊急のものかを各読み出し要求ごとに決定する。所定時間内に要求のサービスを行えなかったことによりビデオデータストリームの再生に問題が生じる場合には、該要求は緊急のものである。また、同様の場合に問題が生じないのであれば、該要求は非緊急のものである。好適には、メッセージが緊急であるか非緊急であるかは、ビデオデータストリームの現在の状態によって決定される。例えば、ビデオデータストリームが現在一時停止されており、要求がその再生を再開させるものである場合には、その要求は非緊急のものである。しかしながら、ビデオデータストリームが再生状態にある場合の要求は緊急のものである。優先順位法は、次いで、各緊急メッセージ毎の最終デッドラインを計算する。優先順位法は、次いで、如何なる緊急メッセージもその最終デッドラインを逸することなく非緊急の要求のサービスを行うのに十分な時間が存在するか否かを判定する。該条件が満たされる場合には、システムは非緊急の要求を取り扱い、該条件が満たされない場合には、緊急の要求が次に処理される。

【００１９】本発明の別の側面によれば、サーバシステム及び方法は、ディスクロード平衡化法を用いて、特定のビデオデータストリームの再生の開始のスケジューリングを行う。該ディスクロード平衡化法は、複数の時間帯を規定する。ここで、好適には、該時間帯の数は記憶装置の数に対応する。該ディスクロード平衡化法は、各

ビデオデータストリームを各時間帯に１つずつ割り当てて、ビデオデータストリームの処理を分配する。該ディスクロード平衡化法は、かかる割り当てを、先ず、ビデオデータストリームを提供することになる記憶装置を識別し、次いで、その記憶装置によりサービスされることになる次に「利用可能な」時間帯を判定することにより行う。更なるビデオデータストリームを扱う容量（帯域幅）を有する場合には、該時間帯は「利用可能な」ものとみなされる。該ディスクロード平衡化法は次いで、その「利用可能な」時間帯を新規に開始されるビデオデータストリームに割り当てる。

【００２０】本発明の更に別の側面によれば、サーバシステム及び方法は、別個のディスクからなる冗長ディスクアレイ（ＲＡＩＤ）システム及び方法を用いてビデオオブジェクトの記憶を行う。該ＲＡＩＤシステム及び方法は、ビデオオブジェクトを複数のデータブロックへと分割し、それらのデータブロックを、ストライピング処理を用いて（ストライプ化された構成で）複数の記憶装置にわたって（即ち、Ｎ個の記憶装置にわたって）記憶する。該システム及び方法によれば、冗長ファクタ（Ｍ）が選択される。該冗長ファクタＭは、信頼性、及びシステム動作中の故障モードでのサービス時間を決定する。Ｍは、Ｎ未満の整数となるように選択される。本発明のこの側面によれば、Ｍ個のデータブロックが記憶される度に、エラー回復ブロックが計算される。好適には、該エラー回復ブロックは、該Ｍ個のデータブロックについて排他ＯＲ演算を行うことにより生成されたパリティコードとなる。Ｍに比べてＮが大きいと、ディスクの故障に遭遇した際に、エラー回復処理が、冗長ファクタＭによって、必要とされる記憶装置のアクセスコールの数に好適に制限される。一実施例では、エラー回復ブロックは、データブロックとインターリーブされて記憶されるが、それに関連するデータは、その記憶装置とは異なる記憶装置に記憶される。かかる本発明の特徴は、ビデオデータ以外のディジタルデータを記憶するためのシステム及び方法に適用可能なものであり、また、サーバシステム以外の記憶システムにも適用可能なものである、ということが理解されよう。

【００２１】本発明の更に別の側面によれば、中央制御モジュール、分配モジュール、及び記憶モジュールは、システムの柔軟性及び拡張性を向上させるためにラックマウント・・・・・ ・・・・・・・・システムにおけるラックマウント・・・・・ ・・・・・・・・・にもそれぞれ適用可能である。

【００２２】本明細書に記載の特徴及び利点は、その全てを含むものではなく、当業者であれば、図面、発明の詳細な説明及び特許請求の範囲を参照することにより、更に多くの特徴及び利点が自明であろう。更に、本書で使用する用語は、基本的に、読み易さ及び本発明の例示を目的として選択されたものであって、本発明の要旨に制限を加えるために選択されたものではなく、かかる本発明の要旨は特許請求の範囲に基づいて決定されるべきである、ということに留意されたい。

【００２３】

【発明の実施の形態】図１は、本発明によるビデオオンデマンド（ＶＯＤ）システム・・・を示すブロック図である。ＶＯＤシステム・・・は、制御入力ソース・・・及びビデオサーバ・・・を備えている。ビデオサーバ・・・は、１つ又は２つ以上の中央制御モジュール・・・と、１つ又は２つ以上の分配モジュール・・・と、１つ又は２つ以上の記憶モジュール・・・とを備えている。該ＶＯＤシステム・・・が調整可能及び拡張可能なものであるため、特定用途で使用される中央制御モジュール・・・、分配モジュール・・・、及び記憶モジュール・・・の数は、分配すべきストリームの数や特定用途におけるビデオ記憶要件等のファクタによって決まる。一好適実施例では、ビデオサーバ・・・は、１つの中央制御モジュール・・・と、１つの分配モジュール・・・と、１つの記憶モジュール・・・とを備えている。更に、調整性及びシステムの拡張性を促進させるために、ビデオサーバ・・・は好適にはラックマウントシステムとする。この場合、各サブコンポーネント（中央制御モジュール・・・、分配モジュール・・・、及び記憶モジュール・・・）はラックマウント用に適応される。

【００２４】制御入力ソース・・・は、記憶されているビデオ情報（ビデオデータ）の取り出し及び表示を制御するための制御信号を生成する任意の入力ソースである。典型的な制御入力ソース・・・は、キーボード、遠隔制御装置、マウス、完全なコンピュータシステム、又はビデオサーバ・・・に連結されたクライアントコンピュータのネットワークを備えるものとなる。本好適実施例では、制御入力ソース・・・は、ビデオサーバ・・・に連結されたビデオクライアント・・・のネットワークである。各ビデオクライアント・・・は、ビデオ制御信号を生成するコンピュータである。従って、ビデオクライアント・・・は、ビデオサーバ・・・に接続されたビデオ要求信号及び制御信号を生成することによって、ＶＯＤシステム・・・により提供される複数のビデオのうちの１つのビデオの選択及びその再生の制御を行うために使用される。ビデオクライアント・・・は、好適にはイーサネットネットワークを使用してビデオサーバ・・・に接続される。しかしながら、本発明によれば、ビデオクライアント・・・をビデオサーバ・・・に接続する他の手段を使用することも可能である、ということが理解されよう。例えば、ビデオクライアント・・・は、ローカルエリアネットワーク、無線通信リンク、光リンク、又は他の通信手段を使用してビデオサーバ・・・に接続することが可能である。

【００２５】ここで再び図１を参照する。記憶モジュール・・・は、１つ又は２つ以上の記憶装置・・・を備えている。該記憶装置・・・の各々は、好適には、従来のハードディスクドライブ、ＣＤ・ＲＯＭドライブ、又はテープドライブ等の大容量記憶装置である。この好適実施例で

は、記憶装置･･･は、･･･････ ････によって製造された大容量（４～９･････）ディスクドライブである。記憶モジュール･･･は、複数のビデオオブジェクト（ビデオシーケンス）を記憶する。一実施例では、ビデオオブジェクトの各々は長編のビデオ映画である。別の実施例では、ビデオオブジェクトは別の形態のビデオコンテンツである。ここで、「ビデオ」という用語は、音声部分及び画像部分を有するコンテンツ、又は音声のみのコンテンツ、又は画像のみのコンテンツ、又はその他の種類のディジタルコンテンツを含むものである、ということが理解されよう。従って、用語「ビデオ」は、ディジタル音楽記録、音声記録、無声画像セグメント等も含むものである。

【００２６】本発明の好適実施例は、「ストライピング」を用いた本発明のＲＡＩＤ技術に従って各ビデオオブジェクトを記憶するものであり、これについては後述する。

【００２７】ストライピングを用いて、各ビデオオブジェクトが複数の「ビデオストライプ」へと分割され、それら各ビデオストライプが異なる記憶装置･･･に記憶される。更に、各ビデオストライプが「データブロック」と呼ばれる複数の･･･････のデータチャンク（･････塊）へと副分割される。

【００２８】中央制御モジュール･･･は、重いマルチスレッドのオペレーティングシステム（好適には･･･ ･････････のオペレーティングシステム･･･････）をＣＰＵ･･･（好適には･････ ･･･････････により製造された･･････プロセッサ）上で実行する高性能パーソナルコンピュータ用マザーボードである。該マザーボードは、･･･････ ･･･････ ････により製造されたものであり、･･･ ･･･ ･･･････ ･･･････により製造されたラックマウント式シャシにマウントされる。該マザーボードはまた、ＳＣＳＩコントローラ及びイーサネットコントローラ等の周辺装置との接続のためのペリフェラルコントロールインターフェイス（ＰＣＩ）バスを備えている。

【００２９】各中央制御モジュール･･･は、中央制御モジュール･･･と記憶モジュール･･･との間、及び中央制御モジュール･･･と分配モジュール･･･との間との通信を促進させるためにイニシエータ･･･････を備えている。イニシエータ･･･････は、･･･････ ･･･････････････ ･････により製造された従来のＳＣＳＩコントローラカードであり、ＰＣＩバスを用いてＣＰＵ･･･に接続される。メモリバッファ･･･は、ＣＰＵ･･･に直接接続されたダイナミックランダムアクセスメモリ（ＤＲＡＭ････図２参照･）内に割り当てられたメモリ空間である。好適には、メモリバッファ･･･はそれぞれ･･･････のメモリであり、従って、各メモリバッファ･･･は１データブロック全体を記憶する大きさとなっている。

【００３０】また、分配モジュール･･･は好適には、･･･ ･･･････ ･･･････により製造された高性能パーソナルコンピュータ用マザーボードである。該マザーボードは、･････ ･･･････ ･･･････により製造されたラックマウント式シャシにマウントされる。該マザーボードは更に、従来のペリフェラルコントロールインターフェイス（ＰＣＩ）バスを備えている。各分配モジュール･･･は、ターゲット･･･、ＣＰＵ･･･、複数のビデオプロセッサ･･･、及びメモリバッファ･･･を備えている。ＣＰＵ･･･は、好適には、･････ ･･･････により製造された･･･････プロセッサである。ターゲット･･･は、････ ･･･････ ･･･････により製造されたモデル･･･････ＳＣＳＩコントローラ等の従来の「ターゲットモードが可能な」ＳＣＳＩコントローラカードであり、ＰＣＩバスを用いてＣＰＵ･･･に接続される。ここで、「ターゲットモードが可能な」とは、イニシエータモードで動作しているＳＣＳＩコントローラからデータを受信するためにターゲットモードで動作するように適応させることが可能なことを意味している。中央制御モジュール･･･と分配モジュール･･･とのインターフェイスをとるために従来のＳＣＳＩコントローラカードを使用することにより、中央制御モジュール･･･が従来のディスクドライブに対して書込みを行っている際に、その中央制御モジュール･･･が分配モジュール･･･にデータを書き込むことが可能になり、これにより、システムのコスト及び複雑さが低減され、システムの信頼性が向上する。

【００３１】ビデオプロセッサ･･･は、ＣＰＵ･･･の制御の下でメモリバッファ･･･から（ビデオストリームを形成する）ビデオデータを受信し、次いで、クライアント･･･に分配するために各ビデオストリームを処理する。一好適実施例では、ビデオプロセッサ･･･は、･････ ･･･ ･･･････ ･････ ･･･････により製造された従来のＭＰＥＧ－１デコーダや･････ ･･･････ ･･･ ･･･ ･･･････により製造された従来のＭＰＥＧ－２デコーダ等の従来のモーションピクチャエキスパートグループ（ＭＰＥＧ）デコーダである。ＭＰＥＧ－１デコーダ又はＭＰＥＧ－２デコーダの選択は、記憶モジュール･･･に記憶されるビデオデータの圧縮に使用される圧縮技術によって決定される。

【００３２】本発明の一好適実施例は、分配モジュール･･･に･･個のビデオプロセッサ･･･を有している。好適には、各ビデオプロセッサ･･･が、それぞれ１つのビデオストリームについて動作する。更に、該好適実施例では、各ビデオプロセッサ･･･の出力は、ＮＴＳＣ規格及びＰＡＬ規格の何れにも互換性を有する（クライアント･･･の）ビデオモニタに直接接続するためのＮＴＳＣ／ＰＡＬ複合信号である。

【００３３】別の実施例では、ビデオプロセッサ･･･はＭＰＥＧ解凍は行わないが、その代わりに他の種類の解凍を行う。更に別の実施例では、ビデオプロセッサ･･･は、イーサネット、ＡＴＭ、ＰＳＴＮネットワーク等の

ネットワークとのインターフェイスをとるように、又は別のクライアント分配手段とのインターフェイスをとるように、ビデオストリームの処理を行う。これらの実施例において、もし必要であれば、ビデオ解凍は、クライアントのロケーションにおいて分配モジュール･･･上で、又はビデオストリーム経路に沿った別のポイントで行われる。

【００３４】中央制御モジュール･･･は、ＳＣＳＩバス･･･により記憶モジュール･･･と接続される。同様に、中央制御モジュール･･･は、ＳＣＳＩバス･･･により各分配モジュール･･･と接続される。該ＳＣＳＩ通信は、中央制御モジュール･･･に配設されたイニシエータ･･･････、並びに記憶モジュール･･･及び分配モジュール･･･の一部をなす対応するＳＣＳＩコントローラ（分配モジュール･･･ではターゲット･･･、記憶モジュール･･･ではＳＣＳＩ回路（図示せず））によって操作される。記憶モジュール･･･及び分配モジュール･･･上のＳＣＳＩコントローラは、「ターゲット」モードで動作する。分配モジュール･･･のＳＣＳＩインターフェイスは、コスト効率の良いインターフェイス機構であり、これにより、各中央制御モジュール･･･が、あたかもハードディスクドライブ又はその他の従来のＳＣＳＩ互換装置にデータを書き込んでいるかのように分配モジュール･･･にデータを分配することが可能になる。

【００３５】本好適実施例では、記憶モジュール･･･との通信を行うために単一のイニシエータ･･･を使用したが、別の実施例では、ＶＯＤシステム･･･において一層多数の記憶モジュール･･･が使用される場合のインターフェイス要件を満たすように、複数のイニシエータ･･･を用いることが可能である。同様に、本好適実施例では、分配モジュール･･･との通信を行うために単一のイニシエータ･･･を使用したが、別の実施例では、ＶＯＤシステム･･･において一層多数の分配モジュール･･･が使用される場合のインターフェイス要件を満たすように複数のイニシエータ･･･を用いることが可能である。

【００３６】本好適実施例では単一の中央制御モジュール･･･を使用したが、本発明の原理は、多数の中央制御モジュール･･･を備えたＶＯＤシステム･･･にも適用することができる。ビデオサーバ･･･中に多数の中央制御モジュール･･･を配設することにより、冗長動作を行うようにＶＯＤシステム･･･を構成することが可能となり、これにより、システムの信頼性及び故障に対する許容性が改善される。更に、多数の中央制御モジュール･･･を備えた構成はシステムの帯域幅を増大させるものとなり、これにより、ＶＯＤシステム･･･によって生成される最大ビデオストリーム数が増大する。

【００３７】好適なシステム構成の１つは、９個の分配モジュール･･･のサービスを行う単一の中央制御モジュール･･･を備えたものとなり、この場合、各分配モジュール･･･は、･･個のビデオプロセッサ･･･を有している。従って、この好適な構成は、最大･･･のビデオストリームを同時に生成するものとなる。別の構成では、･･個ではなく８個のビデオプロセッサ･･･を用いており、従って、最大･･のビデオストリームを分配する。

【００３８】各中央制御モジュール･･･は、１つ又は２つ以上のクライアント･･･からのビデオ制御コマンドを受信して処理する。このビデオ制御コマンドには、例えば、再生、記憶、一時停止、早送り、巻戻し、ビデオ選択等が含まれる。より詳細に言えば、中央制御モジュール･･･上のＣＰＵ･･･は、受信したビデオ制御コマンドをデコードし、そのデコードされたコマンドを実行するために記憶モジュール１３０及び分配モジュール１２０を制御する。中央制御モジュール･･･は、メモリバッファ･･･に対して出入りするビデオデータの非同期の伝送の管理及びスケジューリング等の機能を行う。

【００３９】従来、ビデオサーバシステム（即ち、ビデオオンデマンドシステム）は、２つのカテゴリー、即ち、ストリーミングシステム及び非ストリーミングシステムのうちの１つに含まれる。ストリーミングシステムは、再生要求に応じて明らかに連続するビデオストリームを分配し、これは、再生を変更する（即ち、一時停止や停止等）ための別のユーザコマンドが受信されるまで、またはファイルの終わりに達するまで行われる。また、非ストリーミングシステムでは、ビデオサーバは、進行中のビデオストリームを分配することはなく、その代わりにクライアントの要求に応じてビデオチャンク又はビデオセグメントを分配する。好適には、クライアント･･･からの要求は、ユーザのための明らかに連続する「リアルタイムの」ビデオストリームを生成するように十分頻繁に発生するものであり十分に迅速なものでなければならない。ＶＯＤシステムの好適実施態様は、ストリーミングタイプのビデオサーバである。ストリーミングタイプのビデオサーバは、非ストリーミングタイプのビデオサーバと比べてクライアント･･･とビデオサーバ･･･との間で必要となる対話が少ないという利点を有するものである。従って、ストリーミングタイプのビデオサーバは、エラーの生じる傾向が低く、多数のチャネルに適応することができ、及びクライアント･･･で必要とする複雑性が低いものとなる。

【００４０】ＶＯＤシステム･･･は、多数のバッファ機構を使用してリアルタイムのビデオストリームを分配する。ＣＰＵ･･･の制御下で、データが記憶モジュール･･･からメモリバッファ･･･へ（好適には･･･････のチャンクで）伝送される。このデータが次いで、やはりＣＰＵ･･･の制御下で一層小さなチャンク（好適には･･･････）で分配モジュール･･･のメモリバッファ･･･へと伝送される。ここで、ＣＰＵ･･･の制御下でデータが一層小さなチャンク（好適には･･･････）で各ビデオプロセッサ･･･へ伝送される。各ビデオプロセッサ･･･は、･･･････のデータチャンクを処理して、クライアントロケーションに

分配するためのビデオストリームを生成する。

【００４１】好適には、記憶モジュール･･･と中央制御モジュール･･･との間、及び中央制御モジュール･･･と分配モジュール･･･との間のデータ伝送は、高速のメモリ伝送を得るために、及び該伝送中におけるＣＰＵを用いた動作を回避するために、ダイレクトメモリアクセス（ＤＭＡ）伝送モードを用いて実行される。

【００４２】好適には、分配モジュール･･･は、ターゲットモードで（ターゲットモードでＳＣＳＩインターフェイスを使用して）中央制御モジュール･･･とのインターフェイスをとるので、ビデオデータ及びそれに伴う制御コマンドがアドレススキームを用いて分配モジュール･･･に送られる。各ビデオストリームは、分配モジュール･･･上の指定されたアドレス範囲に割り当てられる。従って、中央制御モジュール･･･が特定のビデオストリームについてビデオデータの書き込みを行っている場合には、特定のデータストリームを固有に指定するために分配モジュール･･･上の宛先アドレスが使用される。同様に、各ビデオストリームに関する進行、エンコード終了、及び一時停止等の制御情報は、特定のビデオストリームに対して各々マッピングされた特定の事前指定されたアドレスに書き込まれる。各々のビデオストリーム及びそれに関連する制御情報のアドレスマッピングは、予め規定される。代替的には、各ビデオストリームのデータと各ビデオストリームに関する制御情報とのマッピングを行うアドレスマップが、システムのスタートアップ中に分配モジュール･･･から受信され、次いで中央制御モジュール･･･で記憶される。

【００４３】図２は、本発明による中央制御モジュール･･･を示すブロック図である。複数のクライアント･･･から受信される制御コマンドのサービスを行うために、中央制御モジュール･･･は、ＣＰＵ･･･に接続されたＤＲＡＭ･･･に記憶されているプログラムコード･･･を使用してマルチタスク処理を行う。ＤＲＡＭ･･･はまた、メモリバッファ･･･（図１にも示されている）を形成する。ＤＲＡＭ･･･は、中央制御モジュール･･･に含まれる従来のコンピュータ用マザーボード上に配設されたメモリ拡張スロットに取り付けられたＤＲＡＭである。プログラムコード･･･は、ＣＰＵ･･･によって実行されるマルチプロセシングスレッド･･･～･･･を含んでいる。マルチプロセシングスレッド･･･～･･･は、リモートプロシージャコール（ＲＰＣ）スレッド･･･、コールバックスレッド･･･、ストリームスレッド･･･、記憶スレッド･･･、及びファイルスレッド･･･を備えている。各スレッドは、ＣＰＵ･･･により実行されるコンピュータプログラムを介したアクティブパスである。

【００４４】図２において、中央制御モジュール･･･はまた、該中央制御モジュール･･･に固有のシステムハードディスク･･･を備えている。該システムハードディスク･･･は、ＤＲＡＭ･･･にロードするためのプログラムコード･･･を記憶する。該システムハードディスク･･･は、更に、サーバコンフィギュレーションファイル･･･、及びビデオカタログサブディレクトリ･･･を記憶する。

【００４５】図３は、マルチプロセシングスレッド･･･～･･･間の関係を示す状態図である。マルチプロセシングスレッド･･･～･･･は、クライアント･･･により要求された際に多数のビデオストリームを再生し及び様々な制御コマンド（即ち、一時停止、停止、巻戻し等）を実行するために、（クライアント･･･上で実行される）クライアントプログラム･･･により生成されるファンクションコールを受信して処理する。

【００４６】リモートプロシージャコール（ＲＰＣ）スレッド･･･は、クライアントプログラム･･･に対してアプリケーションプログラムインターフェイス（ＡＰＩ）を提供し、よって、クライアントプログラム･･･から受ける制御入力（ファンクションコール）の受信を操作する。中央制御モジュール･･･は、ビデオサーバ･･･とクライアント･･･との間のインターフェイスを管理するために、単一のＲＰＣスレッド･･･を生成（実行）する。

【００４７】中央制御モジュール･･･は、各出力ビデオストリーム毎にストリームスレッド･･･を（ＣＰＵ･･･上で）生成し実行する。各ストリームスレッド･･･は、１つのビデオストリームの再生を管理する。

【００４８】コールバックスレッド･･･は、ＣＰＵ･･･によって実行され、ストリームスレッド･･･により生成されるメッセージを操作する。該メッセージは、「ファイルの終わり」またはエラー状態が生じた結果として生成されるものである。

【００４９】ファイルスレッド･･･は、ＣＰＵ･･･によって実行され、ビデオオブジェクトの作成、削除、書き込み、及び読み出しを含むファイル管理を操作する。中央制御モジュール･･･は、多数のファイルスレッド･･･を有している。

【００５０】各記憶装置･･･は、１つ又は２つ以上の記憶スレッド･･･によって管理される。記憶スレッド･･･は、ストリームスレッド･･･、ファイルスレッド･･･、及びＲＰＣスレッド･･･からのメッセージ要求を受信し、次いで適当なディスクアクセス機能及びデータ取り出し機能を行うことにより、そのメッセージ要求のサービスを行う。所与の記憶装置･･･を管理する記憶スレッド･･･の数は、サーバコンフィギュレーションファイル･･･で指定される。好適には、２つの記憶スレッド･･･が各記憶装置･･･を管理する。

【００５１】ここで、図２を再び参照する。各記憶装置･･･は、関連するメッセージ待ち行列･･･をそれぞれ１つずつ有している。該メッセージ待ち行列･･･は、ディスク入出力要求メッセージを記憶するためのファーストインファーストアウト（ＦＩＦＯ）メッセージパイプ（待ち行列）である。ストリームスレッド･･･は、特定の記憶装置･･･からのビデオデータの読み出しを必要とする

場合に、（ディスク入出力を要求する）（ディスクアクセス）メッセージを、適当な記憶装置・・・に対応するメッセージ待ち行列・・・に送る。各メッセージは、該メッセージを生成したストリームスレッド・・・により算出されたデッドラインフィールドを含んでいる。

【００５２】図４は、記憶装置へのアクセス時に使用されるデータ構造及びプログラムモジュール・・・を示すフローチャートである。プログラムコード・・・は、１組のリンクリスト・・・・・・・・・・・データ構造・・・を含んでいる。該リンクリストデータ構造・・・は、自由リスト・・・及び要求リスト・・・を有している。１つの自由リスト・・・及び１つの要求リスト・・・が各記憶装置・・・毎に作成される。自由リスト・・・は、自由メッセージ記憶要素の未ソートのリンクリストであり、要求リスト・・・は、各メッセージに関するデッドラインフィールドに従ってソートされたメッセージのリンクリストである。各記憶スレッド・・・は、最初に自由リスト・・・から記憶要素を取り出すことにより、メッセージの処理を行う。記憶スレッド・・・は、次にメッセージ待ち行列・・・からメッセージを取り出し、その取り出したメッセージを記憶要素に記憶させる。記憶スレッド・・・は次いでそのメッセージを、それに関するデッドラインフィールドに従って要求リスト・・・中にリンクさせる。

【００５３】図５は、本発明による要求リスト・・・を示す説明図である。要求リスト・・・は、その要求リスト・・・の前端がゼロのデッドラインのメッセージ・・・を有するように構成されたメッセージ・・・のリンクリストである。ゼロのデッドラインのメッセージ・・・の後に非ゼロのデッドラインのメッセージ・・・が記憶され、この非ゼロのデッドラインのメッセージ・・・が緊急時に伝わり、緊急性の最も低い非ゼロのデッドラインのメッセージ・・・が要求リスト・・・の後端で共有されるようになる。

【００５４】要求リスト・・・及び自由リスト・・・は共に相互排他ロック・・・を有しており、これにより、要求リスト・・・及び自由リスト・・・へのアクセスが連続的なものとなる。相互排他ロック・・・は、オペレーティングシステムにより提供される従来のロック機構である。

【００５５】処理スレッドの説明

ここで再び図３を参照する。中央制御モジュール・・・は、ＲＰＣスレッド・・・がクライアントプログラム・・・から・・・・・・・・・・・・コールを受信するまでアイドル状態となる。・・・・・・・・・・・・コールは、再生用の新たなビデオストリームをオープンするための要求である。ＲＰＣスレッド・・・は、・・・・・・・・・・・・コールを受信すると、ストリームスレッド・・・に・・・・・・・・・・メッセージを送る。次いで、ストリームスレッド・・・は、オープンされたビデオストリームの再生を操作する。

【００５６】・・・・・・・・・・メッセージを操作する際に、ストリームスレッド・・・は、再生すべきビデオオブジェクトの最初の３つのデータブロックを記憶する記憶装置・・・に対応する３つの記憶スレッドのメッセージ待ち行列・・・の各々に・・・・・・・・・メッセージを送る。好適実施例では、３つのメモリバッファ・・・が各再生ストリーム用にリザーブされ、従って、・・・・・・・・・・メッセージのサービスを行うことによって、新たにオープンされた再生ストリームに関連するメモリバッファ・・・が満たされることになる。

【００５７】各記憶スレッド・・・は、そのメッセージ待ち行列・・・から・・・・・・・・・メッセージを非同期で取り出し、そのメッセージを処理用に優先順位付けする。最終的に処理されると、記憶スレッド・・・は、要求されたデータブロック（好適なブロックサイズは・・・・・・・）を特定のディスクから読み出して、割り当てられたメモリバッファ・・・に該データブロックを書き込むことにより、・・・・・・・・・メッセージを処理する。・・・・・・・・・メッセージのサービスを行った後、記憶スレッド・・・は、前記・・・・・・・メッセージを発したストリームスレッド・・・に・・・・・・・メッセージを送る。

【００５８】次いで、記憶スレッド・・・は、そのメッセージ待ち行列・・・における次に最も時間的に決定的な（時間クリティカル）メッセージを処理する。しかしながら、そのメッセージ待ち行列・・・が空である場合には、記憶スレッド・・・は、そのメッセージ待ち行列・・・にメッセージが送られてくるまでアイドル状態となる。

【００５９】図６は、図３に示したストリームスレッド・・・の状態図である。ストリームスレッド・・・は、・・・・・・・・メッセージを受信するまで・・・・状態・・・となる。

【００６０】メッセージ待ち行列に・・・・・・・・・メッセージを送った後、ストリームスレッド・・・は、・・・・・・・状態・・・に移行する。ストリームスレッド・・・は、・・・・・・・状態・・・にあるとき、・・・・・・・・・メッセージが送られた各記憶スレッド・・・から・・・・・・・・・メッセージを受信するまで待機する。記憶スレッド・・・により送られた・・・・・・・・・メッセージは、記憶スレッド・・・が・・・・・・・・・要求を処理したことを示すものである。・・・・・・・・・メッセージが受信されると、ストリームスレッド・・・は・・・・・・状態・・・に移行する。

【００６１】ここで再び図３を参照する。ＲＰＣスレッド・・・は、クライアントプログラム・・・から・・・・・・・・・・・・コールを非同期で受信する。次いで、ＲＰＣスレッド・・・は、・・・・・・・・・・メッセージをストリームスレッド・・・に送る。次いで、ストリームスレッド・・・は、該ストリームの再生を操作する。

【００６２】ここで再び図６を参照する。ストリームスレッド・・・が・・・・・・状態・・・にあるとき、該ストリームスレッド・・・は、ＲＰＣスレッド・・・から・・・・・・・・・・メッセージが受信されるまで待機する。ストリームスレッド・・・は、好適には以下で説明するスケジューリングプロトコルに従って、該ストリーム用の開始時間帯を選択することにより、・・・・・・・・・・メッセージを操作する。開始時

間帯の選択後、メモリバッファ･･･からビデオデータの第1のサブブロック（･･･････）を取り出し、宛先出力ポートを含む分配モジュール･･･に該サブブロックを送ることにより、再生が開始される。サブブロックの送信後、ストリームスレッド･･･は、･･･････状態･･･に移行する。

【0063】･･･････状態･･･にあるとき、ストリームスレッド･･･は、ＲＰＣスレッド･･･又は記憶スレッド･･･の何れかから新たなメッセージが到着したか否かを判定し、受信したメッセージを処理する。受信する可能性のあるメッセージとしては、･･･････････メッセージ、･･･････････メッセージ、及び･･･････････メッセージが挙げられる。各々のメッセージは次のように操作される。

【0064】･･･････････････メッセージがＲＰＣスレッド･･･から送られた場合、ストリームスレッド･･･は、･････状態･･･に移行する。

【0065】･･･････････････メッセージがＲＰＣスレッド･･･から送られた場合、ストリームスレッド･･･は、まだ分配モジュール･･･に送られていないメモリバッファ･･･中のデータブロックを破棄する。次に、新たな位置へのジャンプによって取り出されたビデオデータ（データブロック）を記憶するために、ストリームスレッド･･･による使用のために割り当てられていなかったメモリバッファ･･･が記憶スレッド･･･による使用のために割り当てられる。･･･････････メッセージの処理後、ストリームスレッド･･･は、･･･････状態･･･でループして次のメッセージの受信を待つ。

【0066】･･･････････････メッセージが記憶スレッド･･･から送られた場合、及び、･･･････････メッセージがエラーを伴うことなく操作されたことを･･･････････メッセージが示す場合には、ストリームスレッド･･･は、対応するメモリバッファ･･･をレディ状態にマークし、次いで･･･････状態･･･でループする。

【0067】････ ･･･････････メッセージが記憶スレッド･･･から送られた場合、及び、･･･････････メッセージがエラーに遭遇したことを･･･････････メッセージが示す場合には、ストリームスレッド･･･は、コールバックスレッド･･･に･･･････････････メッセージを送り、･････状態･･･に移行する。次いで、コールバックスレッド･･･は、･･･････････････メッセージの受信時に、ビデオコマンドを発したクライアントプログラム･･･にコールバックを行って、ビデオストリーム中でエラーに遭遇したことをクライアントプログラム･･･に知らせる。

【0068】･･･････状態･･･では、ストリームスレッド･･･は、等時性のビデオストリームを維持するためにタイマによって更に制御される。「等時性」とは、非破裂性又は「ほぼ一定速度」を意味するものである。等時性のビデオストリームを維持するために、各々が･･･････のデータサブブロックが所定時間内に分配モジュール･･･に送られる。各々のデータサブブロックを分配モジュール･･･に送る際に、ストリームスレッド･･･は、該データサブブロックがメモリバッファ･･･中の最後のサブブロックであったか否かを判定する。該データサブブロックが最後のサブブロックであった場合には、ストリームスレッド･･･は、メモリバッファ･･･を「利用可能」とマークし、記憶装置･･･からの更なるビデオデータ（･･･････のデータブロック）の取り出しを開始させるために適当な記憶スレッド･･･に･･･････････メッセージを送る。ストリームスレッド･･･は更に、ビデオファイルの終わりに達したか否かを判定する。ビデオファイルの終わりに遭遇した場合には、ストリームスレッド･･･は、コールバックスレッド･･･に･･･････････････メッセージを送り、････状態･･･に移行する。次いで、コールバックスレッド･･･は、ビデオコマンドを発したクライアントプログラム･･･にコールバックを送って、ビデオストリームが正常終了したことをクライアントプログラム･･･に知らせる。しかしながら、ビデオファイルの終わりに達していなかった場合には、ストリームスレッド･･･は、･･･････状態･･･でループする。

【0069】････状態･･･にあるとき、ストリームスレッド･･･は、ＲＰＣスレッド･･･から受信したメッセージを処理する。（クライアントプログラム･･･から･･････････コールを受信した結果として）ＲＰＣスレッド･･･から･･･････････メッセージが送られた場合には、ストリームスレッド･･･は、記憶されているビデオファイル上の新たなジャンプ先位置からビデオデータを取り出すために、メモリバッファ･･･のアドレスを記憶スレッド･･･に送る。メモリバッファ･･･のアドレスの送信後、ストリームスレッド･･･は･･･････状態･･･に移行する。（クライアントプログラム･･･からの･･･････････コールの結果として）ＲＰＣスレッド･･･により･･･････メッセージが送られた場合には、ストリームスレッド･･･は、該ストリームに関連する分配モジュール･･･にストリームの再生のクローズを知らせるコマンドを送出する。次いでストリームスレッド･･･は････状態･･･に移行する。

【0070】･･･････状態･･･にあるとき、ストリームスレッド･･･は、ＲＰＣスレッド･･･により送られたメッセージを処理する。（クライアントプログラム･･･により送られた･･･････････コールの結果として）ＲＰＣスレッド･･･から･･･････メッセージが送られた場合には、ストリームスレッド･･･は、メモリバッファ･･･中のあやゆるデータを破棄し、ビデオファイル中の新たなジャンプ先位置で始まるビデオデータを取り出すために、その解放されたメモリ空間を対応する記憶スレッド･･･に割り当てる。ストリームスレッド･･･は、次いで･･･････状態･･･に移行する。

【0071】（クライアントプログラム･･･からの･･･････････コールの結果として）ＲＰＣスレッド･･･から･･･････メッセージが送られた場合には、ストリーム

スレッド･･･は、該ストリームに関連する分配モジュール･･･にストリームの再生のクローズを知らせる。次いでストリームスレッド･･･は････状態･･･に移行する。

【0072】（クライアントプログラム･･･からの･･･････コールの結果として）ＲＰＣスレッド･･･から･･･････メッセージが送られた場合には、ストリームスレッド･･･は、当該ビデオストリームに関する開始時間スロットを選択し、該開始時間スロットに達した後、ビデオディスクの･･･････のカレントブロックを、（中央制御モジュール･･･上の）メモリバッファ･･･から、該ビデオストリームのための宛先ポートを含む分配モジュール･･･に送る。ストリームスレッド･･･は、次いで･･･････状態･･･に移行する。

【0073】･････状態･･･にあるとき、ストリームスレッド･･･は、ＲＰＣスレッドからの･･･････メッセージを処理する。（クライアントプログラム･･･からの･･･････コールの結果として）ＲＰＣスレッド･･･から･･･････メッセージが送られた場合には、ストリームスレッド･･･は、ストリームの再生がクローズされたことを該ストリームに関連する分配モジュール･･･に知らせる。ストリームスレッド･･･は次いで････状態･･･に移行する。

【0074】ストリームスレッドによるメッセージ要求の優先順位付け

ＶＯＤシステム･･･は、優先順位スキームを用いて、多数のストリームスレッド･･･から各記憶スレッド･･･に送られるディスク入出力要求を要求するメッセージの操作のスケジューリングを行う。該優先順位スキームは、好適には、要求を行っている全てのストリームスレッド･･･がそれぞれのビデオストリームの連続的な再生を維持することができるように全てのメッセージを確実に完了させる（操作する）ものとなる。

【0075】該優先順位スキームによれば、各メッセージはそれに関するデッドラインフィールドを有している。ストリームスレッド･･･が、中央制御モジュール･･･のバッファを満たすために、ディスク入出力を要求するメッセージ（･･･････メッセージ）を記憶スレッド･･･に送ると、ストリームスレッド･･･は、該メッセージについてのデッドラインを計算し、その該デッドラインを該メッセージと共に記憶スレッド･･･に送る。該デッドラインは、ストリームスレッド･･･の現在の状態によって決まる。該デッドラインは、０から最大値までの整数である。デッドラインを有さないメッセージには「ゼロ」のデッドライン値が与えられ、デッドラインを有するメッセージにはその緊急性に対応したデッドライン値が割り当てられ、この場合、大きなデッドライン値を有するメッセージほど緊急性が低く、小さなデッドライン値を有するメッセージほど緊急性が高いものである。

【0076】通常の再生時、即ち、･･･････状態･･･にある場合には、デッドラインの計算は、ストリームスレッド･･･による分配モジュール･･･に対するもっとも最近のデータ書き込みに関する開始時間に、ストリームに関する全てのメモリバッファ･･･中のデータ消費時間（即ち、ビデオデータの再生に要する時間）を加えることにより行われる。好適には、該データ消費時間は、各メモリバッファ･･･のサイズに、ビデオストリームに関するメモリバッファ･･･の数を乗算し、次いでその積を出力データ速度で除算すること（即ち、バッファサイズ×バッファ数÷データ速度）により算出される。

【0077】ストリームの再生が開始する前のバッファの初期プライミング時に（即ち、･･･････状態･･･にある際に）及び･･･････状態･･･にある際に、デッドラインがゼロにセットされる。この「ゼロ」は、メッセージが絶対的なデッドラインを有さないこと、及び、かかるサービスによりメッセージ待ち行列･･･中の他のメッセージがそれらのデッドラインを逸さないこととなる場合に該メッセージがサービスされるべきであることを示すものである。

【0078】ストリームスレッド･･･が･･･････状態･･･にあり、該ストリームスレッド･･･により･･･････････メッセージが受信された場合、ストリームスレッド･･･は、該ストリームスレッド･･･に関するメモリバッファ･･･中のデータを破棄する。次いでストリームスレッド･･･は、記憶されているビデオオブジェクト中の新たな（ジャンプ先）位置から取り出されたデータで満たすためにメモリバッファ･･･のアドレスを適当な記憶スレッド･･･に送る。･･･････････メッセージに関するデッドラインは、該メッセージが絶対的なデッドラインを有さないこと、及び、かかるサービスによりメッセージ待ち行列･･･中の他のメッセージがそれらのデッドラインを逸さないこととなる場合に該メッセージがサービスされるべきであることを示す「ゼロ」である。

【0079】ストリームスレッド･･･が通常再生モードにある際、即ち、･･･････状態･･･にある際に、該ストリームスレッド･･･によって･･･････････メッセージが受信された場合、該ストリームスレッド･･･は、該ストリームスレッド･･･に関連すると共に現在の時間に記憶スレッド･･･の応答時間を加算したものよりも遅いデッドラインを有するデータを有するメモリバッファ･･･中のデータを破棄する。次いでストリームスレッド･･･は、新たなビデオ位置（即ち、ビデオファイル中のジャンプ先位置）から取り出されたデータで満たすと共に以前に記憶されていたデータに関連していたデッドラインと同一のデッドラインを維持するために、データが破棄された該メモリバッファ･･･のアドレスを適当な記憶スレッド･･･に送る。

【0080】記憶スレッドの処理

記憶スレッド･･･は、中央制御モジュール･･･のスタートアップ時に作成され、記憶装置･･･に対するアクセスを管理する。ここで再び図４を参照する。各記憶装置･･･

に対するアクセスは、各記憶装置･･･に関連するリンクリスト（要求リスト･･･及び自由リスト･･･）によって制御される。各記憶装置･･･を管理する記憶スレッド･･･の数は、コンフィギュレーションファイル･･･を読み出すことにより決定される。各記憶装置･･･毎に１つ以上の記憶スレッド･･･が作成される場合には、要求リスト･･･及び自由リスト･･･とアクセスするためにロック機構（･･･････････）が用いられる。

【００８１】図７は、各記憶スレッド･･･により行われるメッセージ待ち行列処理･･･を示すフローチャートである。記憶スレッド･･･は、記憶装置･･･に関連する２つ以上の記憶スレッド･･･が存在するか否かを判定することにより処理を開始する。記憶装置･･･に関連する２つ以上の記憶スレッド･･･が存在する場合には、現在の記憶スレッド･･･が、記憶装置･･･に関連する･･･････ ･･･････を得て、リンクリスト･･･（要求リスト･･･及び自由リスト･･･）をロックする（ステップ･･･）。

【００８２】･･･････ ･･･････が固定されると（及びリンクリスト･･･がロックされると）、記憶スレッド･･･がメッセージを処理する。記憶スレッド･･･は、次に自由リスト･･･からメッセージ記憶要素をリムーブする（アンリンクする）。次いで、記憶スレッド･･･は、取り出されたメッセージをアンリンクされたメッセージ記憶要素に記憶させ（ステップ･･･）、該メッセージをそれに関連するデッドラインに従って要求リスト･･･に挿入する（ステップ･･･）。詳細には、挿入されるメッセージ（新たなメッセージ）が非ゼロのデッドラインを有している場合には、記憶スレッド･･･は、要求リスト･･･のサーチをその後端から開始し（即ち、該後端が最も緊急性の低い非ゼロのデッドラインを有している）、新たなメッセージを、該新たなメッセージよりも早いデッドラインを有する最初のメッセージの直後で要求リスト･･･に挿入する。要求リスト･･･中のメッセージで該新たなメッセージよりも早いデッドラインを有するものが１つも存在しない場合には、該新たなメッセージは要求リスト･･･の最初に挿入される。

【００８３】しかしながら、新たなメッセージがゼロのデッドラインを有している場合には、記憶スレッド･･･は、要求リスト･･･のサーチをその前端から開始し（即ち、該前端が最も緊急性の高いデッドラインを有している）、新たなメッセージを、非ゼロのデッドラインを有する最初のメッセージの直前で要求リスト･･･に挿入する。要求リスト･･･中のメッセージで非ゼロのデッドラインを有するものが１つも存在しない場合には、該新たなメッセージは要求リスト･･･の最後に挿入される。新たなメッセージが要求リスト･･･に挿入された後、記憶スレッド･･･は、次いで･･･････ ･･･････を解放してリンクリスト･･･をアンロックする･ステップ････。記憶スレッド･･･は、メッセージ待ち行列･･･が空になるまでメッセージ待ち行列処理･･･を繰り返す。次いで、記憶スレッド･･･は、要求リスト･･･中の優先順位付けされたメッセージの処理へと進む。

【００８４】図８は、要求リスト･･･中の優先順位付けされたメッセージの記憶スレッド･･･による処理･･･を示すフローチャートである。

【００８５】記憶装置･･･に対して２つ以上の記憶スレッド･･･が存在する場合、現在の記憶スレッド･･･は、該記憶装置･･･に関連する･･･････ ･･･････を得て、リンクリストデータ構造･･･（自由リスト･･･及び要求リスト･･･）をロックする（ステップ･･･）。

【００８６】該データ構造をロックした後、記憶スレッド･･･が、非ゼロのデッドラインのメッセージにそれらのデッドラインを逸させることなく要求リスト･･･中のゼロのデッドラインのメッセージをサービスするだけの十分な時間があるか否かを判定する。記憶スレッド･･･は、要求リスト･･･中の非ゼロのデッドラインのメッセージの操作についての最も最近の開始時間を計算する･ステップ････ことによりこの決定を行う。該最も最近の開始時間は、緊急性の最も低い非ゼロのデッドラインを有する要求リスト･･･の終わりから開始して、各メッセージ毎に、以前のメッセージ及び現在のメッセージに関連する該メッセージのデッドラインについて計算された最も最近の開始時間のうち一層小さいものから、期待されるディスクアクセス（ディスク入出力）時間を減算することで最も最近の開始時間を計算することにより、繰り返し計算される。

【００８７】最も最近の開始時間を計算する場合、該最も最近の開始時間は、最初に、最も最近の開始時間により表現可能な最大整数値へと初期化される･ステップ･･･。更に、ディスクアクセス時間は、要求リスト･･･に関連する特定の記憶装置･･･から１つのデータブロック（･･･････のデータ）を読み出すのに要する時間に相当する。

【００８８】次に、記憶スレッド･･･は、ステップ･･･の比較を行い、計算された最も最近の開始時間が与えられた場合にゼロのデッドラインのメッセージを操作するのに十分な時間が存在するか否かを判定する。この判定は、現在の時間と、最も最近の開始時間及び特定の記憶装置･･･からの期待されるディスクアクセス時間（１データブロック（･･･････のデータ）の読み出しに要する時間）の差とを比較することにより行われる（ステップ････。

【００８９】現在の時間が、最も最近の開始時間及び期待されるディスクアクセス時間の差よりも小さい場合には、ゼロのデッドラインのメッセージを操作すると共に最も最近の開始時間の要件を依然として満たすだけの十分な時間が存在する。従って、かかる場合には、要求リスト･･･中の最初のメッセージが処理のためにリムーブされる（ステップ････。この最初のメッセージは、ゼロのデッドラインのメッセージ又は最も緊急性の高い（即

ち最小デッドラインの）メッセージとなる。

【００９０】しかしながら、現在の時間が、最も最近の開始時間及び期待されるディスクアクセス時間の差よりも大きい場合には、ゼロのデッドラインのメッセージを操作すると共に最も最近の開始時間の要件を依然として満たすだけの十分な時間が存在しない。従って、かかる場合には、要求リスト･･･中の最初の非ゼロのデッドラインのメッセージが処理のためにリムーブされる（ステップ･･･。

【００９１】処理のためにメッセージをリムーブした（ステップ･･･又は･･･後、記憶スレッド･･･は、リンクリストデータ構造･･･をアンロックし（ステップ･･･、次いで該メッセージを処理する（ステップ･･･。この処理の後、記憶スレッド･･･は次いでリンクリストデータ構造･･･をロックし（ステップ･･･、ステップ･･･で処理されたメッセージにより占有されたメッセージ記憶要素を自由リスト･･･に挿入する（ステップ･･･。この挿入の後、リンクリストデータ構造･･･はアンロックされる（ステップ･･･。

【００９２】記憶スレッド処理･･･の完了後、記憶スレッド･･･は、記憶スレッド処理･･･が開始してからメッセージ待ち行列･･･に書き込まれたメッセージを取り出すために、図７に示したメッセージ待ち行列処理･･･へと処理を戻す。

【００９３】記憶モジュールのデータ構造及びアクセス機構

ＶＯＤシステム･･･は、ビデオオブジェクトの記憶を管理するためにハイブリッドファイル管理機構を用いる。該ハイブリッドファイル管理機構は、多数の名称付きビデオオブジェクト（即ちビデオファイル）の管理タスクを簡素化すると共に生････ディスクドライブの最大性能の帯域幅を完全に利用するために、中央制御モジュール･･･上で実行されるオペレーティングシステムによって提供されるファイルシステムサービスと生ディスクアクセス法との両方を含むものである。

【００９４】一般に、ビデオオブジェクト自体のサイズは、該ビデオオブジェクトに関する制御情報（例えば、ビデオ属性、作成日時、及び記憶マップ等）に比べて極めて大きなものである。典型的には、ビデオオブジェクトのサイズは････、その制御情報のサイズは････又はそれ未満といったものである。更に、ビデオオブジェクトについての入出力動作の回数はその制御情報についての入出力動作の回数を大きく上回るものである。ＶＯＤシステム･･･は、ビデオオブジェクト自体について記憶及びアクセスを行うための生ディスク法を用いる。このため、オペレーティングシステムのファイルシステムに関連する空間上及び性能上のオーバーヘッドを回避する（バイパスする）ことにより、空間的な要件が最小限となり、性能が最適化される。

【００９５】しかしながら、ＶＯＤシステム･･･は、各ビデオオブジェクトに関連する制御情報を記憶するためにオペレーティングシステムのファイルシステムを使用する。かかるファイルシステムを使用することにより、ビデオオブジェクトの名称空間マッピング･･･････の管理、ディレクトリ情報の維持、及び制御情報のための記憶空間の動的な割り当て及び割り当て解除を行うという複雑性が排除される。その上、ソフトウェアのテスト、システムのメンテナンス、及び将来のアップグレードに対する備えが簡単になる。同時に、記憶スペース及び性能に関するオーバーヘッドにより生じる不利益が最小限となる。これは、ビデオオブジェクトと比較して制御データの方がサイズが比較的小さく、及び入出力要求の数が比較的少ないことによる。

【００９６】ここで再び図２を参照する。中央制御モジュール･･･中のシステムディスク･･･は、ビデオカタログサブディレクトリ･･･及びサーバコンフィギュレーションファイル･･･を含んでいる。

【００９７】ビデオカタログサブディレクトリ･･･はディレクトリ（例えば･･･････････）であり、複数の名称付きファイルを有している。この場合、各々の名称付きファイルは、記憶モジュール･･･に記憶されている同じ名称のビデオオブジェクトに対応するものである。かかる名称付きファイルは、ビデオ属性、再生データ速度、及び同時に発生する最大ユーザ数等の制御情報を含んでいる。

【００９８】また、サーバコンフィギュレーションファイル･･･（例えば･･･････････）は、記憶モジュール･･･における記憶装置･･･の記憶上の割り当てに関する情報を含んでいる。かかる情報には、例えば、生装置名称、ストライプセグメントサイズ、及び冗長性に関する情報等が含まれる。サーバコンフィギュレーションファイル･･･は、システムのスタートアップ時に読み出され、ＶＯＤシステム･･･を構成するために使用される。

【００９９】更に、システムディスク･･･は、記憶モジュール･･･における記憶装置･･･の数と同数の多くのマウントポイントを有している。通常の動作中に、各記憶装置･･･の制御区画が該マウントポイントのうちの１つにマウントされる。

【０１００】ＶＯＤシステム･･･の構成中に、各記憶装置･･･は、２つの区画、即ち制御区画及びデータ区画へとフォーマットされる。

【０１０１】記憶装置･･･のフォーマット中に、各制御区画にファイルシステムが作成される。各制御区画は、対応するデータ区画についてセグメントの利用可能性を指定する自由スペースビットマップを含んでいる。

【０１０２】制御区画はまた、多数の名称付きファイルを含んでおり、該名称付きファイルの各々は、ビデオオブジェクトのストライプのスペースマップを１つずつ含んでいる。１つのスペースマップは、１つのビデオスト

ライブに含まれている……のデータブロックの各々に関するアドレス情報のマッピングを行う。従って、スペースマップは、記憶装置…上のビデオストライプの……のデータブロックの各々の位置を定めるために使用される。一層詳細には、スペースマップは、ビデオオブジェクトのストライプ内の論理ブロック番号を、同じ記憶装置…上のデータ区画内の物理セグメント番号に変換する。スペースマップファイルの名称は、対応するビデオオブジェクトの名称にストライプ番号を付加することにより形成される。

【０１０３】各記憶装置…のデータ区画は、生ディスク区画として形成される（即ち、如何なるオペレーションシステム情報も伴わずにディスクがフォーマットされる）。データ区画に対するアクセス及び記憶管理は、完全に中央制御モジュール…の制御の下で行われる。より詳細には、記憶スレッド…がデータ区画のアクセス及び記憶管理を制御する。

【０１０４】記憶モジュールにおける記憶装置のフォーマッティング

記憶装置…は複数のグループ（ストライプグループと呼ばれる）に編成され、該各グループには番号（ストライプグループ番号と呼ばれる）が１つずつ割り当てられる。１つのビデオオブジェクトが複数のビデオストライプへと分割される際、それらのビデオストライプは特定のストライプグループに割り当てられる。ビデオオブジェクト内の各ビデオストライプは、該割り当てられたストライプグループ内で別個の記憶装置…に記憶される。記憶モジュール…における各記憶装置…は、特にＶＯＤシステム…のためにフォーマットされる。

【０１０５】該フォーマット処理の際に、ユーザは、ストライプグループ番号、ストライプ番号、生装置アドレス、ストライプセグメントサイズ、及びフォーマットすべきディスクに関する一次／二次標識等の記憶情報を指定する。ユーザはまた、例えばストライプグループ２及びストライプ４のディスクを……とするといった、所望の名称付け上の約束を用いてマウントポイントを作成する。

【０１０６】次に、……サーバコンフィギュレーションファイル…がオープンされる。サーバコンフィギュレーションファイル…が存在しない場合には、新たなサーバコンフィギュレーションファイル…が作成される。ユーザにより指定された記憶フォーマット情報は、サーバコンフィギュレーションファイル…に対する妥当性の検査が行われる。該妥当性検査の後、新たなドライブ名及び情報がサーバコンフィギュレーションファイル…に追加される。

【０１０７】次に、ディスクが２つの区画にフォーマットされる。区画０（制御区画）は、マウント可能なものとして規定され、該区画０にファイルシステムが作成される。区画１（データ区画）は、マウント不能なものとして規定される。

【０１０８】次に、区画０が、以前に生成されたマウントポイントにマウントされる。従って、……等のファイルが、自由スペースビットマップとして区画０に作成される。該ファイルは次いで、区画１における全てのセグメントが利用可能である（未割り当てである）ことを示すよう初期化される（但し、セグメント０を除く）。次いで、区画０がマウント解除される。

【０１０９】次に、区画１がオープンされて、ストライプグループ番号、ストライプ番号、ストライプ用のマウントポイント、一次／二次フラグ、活動ディスクフラグ、一次ディスクのための生装置名称、及び二次ディスクのための生装置名称等の情報が、セグメント０に書き込まれる。

【０１１０】セグメント０への書き込み後、区画１及びコンフィギュレーションファイルがクローズされる。

【０１１１】記憶モジュールのスタートアップ処理

記憶装置…のフォーマット後、ＶＯＤシステム…をスタートアップさせることが可能になる。該スタートアップ処理は、サーバコンフィギュレーションファイル……をＤＲＡＭ…に読み込み、次いで、該サーバコンフィギュレーションファイル…中のコンフィギュレーション情報を実際のハードウェア構成と比較することにより妥当性検査を行うことを含む。

【０１１２】サーバコンフィギュレーションファイル…の妥当性検査の後、各ディスクが次の処理によって初期化される。

【０１１３】…　ディスクの制御区画（区画０）をその対応するマウントポイント（例えば、……）にマウントし、……　制御区画からメモリに自由スペースビットマップファイルを読み込み、通常の動作時にスペースの割り当て及び割り当て解除のために該ファイルに対して効率的にアクセス及び更新を行うことができるようにし、……　ディスク上のビデオオブジェクトのストライプに対する後続の通常のアクセスのために該ディスクのデータ区画（区画１）をオープンする。

【０１１４】ビデオオブジェクトのオープン

ＶＯＤシステム…は、スタートアップ処理を完了すると、クライアントプログラム…がビデオオブジェクトを作成するために……ファンクションコールを行うまで待機する。例えば、クライアントプログラム…は、……と呼ばれるビデオオブジェクトを作成するために……ファンクションをコールすることができる（ステップ…。

【０１１５】……コールに応じて、ＶＯＤシステム…は、図９にフローチャートで示すビデオオープン処理を行って、記憶モジュール…上のビデオオブジェクトをオープンする。

【０１１６】該ビデオオープン処理は、ビデオカタログ

ディレクトリ…（例えばディレクトリ……）中のビデオカタログファイル……を作成する（ステップ…ことにより開始する。VODシステム…は次いで、ビデオ属性、データ速度、ビデオ長、及び作成日時等の制御情報を、ビデオカタログファイル……に書き込む（ステップ…。

【0117】次に、外処理は、ストライプグループ内の各記憶装置…毎にスペースマップを生成する（ステップ…。該スペースマップは、特定のビデオストライプの各データブロックを記憶装置…上のアドレスに変換するものである。該スペースマップは、各記憶装置…の制御区画（即ち区画0）に存在する。該スペースマップファイルの名称は、好適には、ストライプの総数及び特定のストライプ番号をビデオオブジェクト名に付加することにより生成される。例えば、ビデオ……に6つのストライプが存在する場合、該ビデオオブジェクトのストライプ3に関連するスペースマップファイルに……という名称を付けることができる。該スペースマップ生成プロセス…は、ビデオオブジェクトの各ストライプ毎に繰り返される。次いで、それらのスペースマップファイルが書き込み動作のためにオープンされる。

【0118】次いで、作成されてオープンされた各スペースマップファイル毎に、VODシステム…は、記憶装置…に対応するファイル制御ブロックチェーンに制御ブロックを挿入する・ステップ…）。各記憶装置…は、ファイル制御ブロックチェーンを1つずつ有している。ファイル制御ブロックチェーンは、一連の制御ブロックであり、DRAM…中で共有される。制御ブロックは、各ビデオストライプに関連する制御情報のコピーであり、特に、記憶装置…の制御区画に記憶されているスペースマップのコピーを含んでいる。ファイル制御ブロックチェーンにおける制御ブロックがDRAM…に記憶されているため、該制御ブロック中のスペースマップは、各制御区画で共有される実際のスペースマップよりも速いアクセス時間を有するものとなる。

【0119】次いで、クライアントプログラム…が、ビデオオブジェクトデータの書き込みのために……ファンクションをコールし・ステップ…）、VODシステム…は、各データブロック毎に、データブロックを記憶するための特定のストライプグループ中の記憶装置…を選択する・ステップ…）。記憶装置…を選択した後、VODシステム…は、利用可能なスペースを求めて対応する自由スペースビットマップをサーチすることにより、データブロックのためのメモリの割り当てを行う・ステップ…）。

【0120】ビデオオブジェクトデータを記憶するためのメモリが割り当てられた後、中央制御モジュール…が、ビデオオブジェクトの各ストライプ毎にファイル制御ブロックを更新し・ステップ…）、また記憶割り当てを反映するよう自由スペースビットマップを更新する・ステップ…）。次に、中央制御モジュール…は、スペースマップに従ってストライプグループ中に存在する各記憶装置…の区画1にビデオオブジェクトデータを書き込むために、生ディスク書き込み動作を生成する・ステップ…）。全てのデータブロックの書き込み後、クライアントプログラム…は、……ファンクションをコールする。該……ファンクションを受信した際に、VODシステム…は、各記憶装置…に記憶されているスペースマップを更新する。

【0121】ビデオオブジェクトの再生

ビデオオブジェクトの再生は、クライアントプログラム…が、……次いで……ファンクションコールを生成することにより開始される。クライアントプログラム…は、例えば、……及び……ファンクションをコールして、……という名称が付けられたビデオオブジェクトの再生を開始させることができる。図10は、再生のためにビデオオブジェクトをオープンする処理を示すフローチャートである。

【0122】……ファンクションがコールされると・ステップ…）、プログラムコード…は、ビデオカタログファイル…（例えば……）をオープンし・ステップ…）、そのコンテンツ（即ち内容）を読み出す。ビデオカタログファイル…から読み出された情報（ストリームデータ速度、ビデオオブジェクトサイズ等）は、ビデオオブジェクトの再生を制御するために使用される。

【0123】次いで、ビデオオブジェクトの各ストライプ毎に、プログラムコード…は、（特定のビデオストライプに割り当てられた記憶装置…に記憶されている）スペースマップファイルを読み出して・ステップ…）、制御ブロックを生成する。次いで、プログラムコード…は、ビデオストライプが割り当てられている記憶装置…に関連する制御ブロックチェーンのサーチを行う・ステップ…）。ビデオストライプに関する制御ブロックが制御ブロックチェーン中に既に存在する場合には、プログラムコード…は使用カウントをインクリメントする・ステップ…）。また、該制御ブロックが制御ブロックチェーン中に既に存在しない場合には、プログラムコード…は、該制御ブロックチェーンに該制御ブロックを追加して・ステップ…）、使用カウントを1にセットする。

【0124】前記サーチ…を行った後、プログラムコード…は、次いで制御ブロックに記憶されているスペースマップ情報を使用して、記憶装置…の区画1に対して生ディスク読み出し動作を行って・ステップ…）、該ビデオオブジェクトデータをメモリバッファ…に読み込む。

【0125】続いてクライアントプログラム…によって……ファンクションがコールされると・ステップ…）、中央制御モジュール…は、ビデオオブジェ

クトデータを、その処理のためにメモリバッファ･･･から分配モジュール･･･へと送る。プログラムコード･･･は、ビデオオブジェクトの終わりに達するまで、又はユーザにより指定された終了条件（例えば時間制限）等の阻止条件が発生するまで、生ディスク読み出し動作を続行する･ステップ･･･）。次いで、プログラムコード･･･は、コールバックファンクションによってクライアントをコールして、再生の終了をクライアントプログラム･･･に知らせる。

【０１２６】次いで、クライアントプログラム･･･は･･････････ファンクションをコールする。プログラムコード･･･は、次いで、･･･････････ファンクションコールに応じてビデオオブジェクトの各ストライプ毎にクローズ処理を行う。

【０１２７】該クローズ処理には、制御ブロックチェーン中のスペースマップファイルに関する使用カウントを判定することが含まれる。該判定後、使用カウントがゼロである場合、制御ブロックチェーンから該制御ブロックが削除される。

【０１２８】該判定後、プログラムコード･･･は、ビデオオブジェクトのストライプに関するスペースマップファイルをクローズする。

【０１２９】最終的に、プログラムコード･･･は、ビデオオブジェクトに関するビデオカタログファイル･･･（例えば･･･････････････）をクローズする。

【０１３０】ディスクロード平衡化（スケジューリング）

マルチストリームＶＯＤシステム･･･において、各ビデオ再生ストリームの開始時間が調整されない場合には、読み出しを要求するあまりに多数のメッセージを同時に受けることによって１つ又は２つ以上の記憶装置･･･がオーバーロードされる可能性がある。かかる事態が発生すると、連続するストリーム再生に関する時間要件を満たすよう適時に操作されないメッセージが生じる可能性がある。これは、ビデオ再生に望ましくない欠陥を生じさせるものとなる。ＶＯＤシステム･･･は、好適には、データストライプスキームを用いて多数の記憶装置･･･へのビデオオブジェクトの記憶をインターリーブさせ、更に、スケジューリング法を用いて各ビデオストリームの開始時間を調整し、これにより複数の記憶装置･･･の何れもオーバーロードされることがないようにする。該スケジューリング法はまた、ストリームの開始前の時間遅延を最小限にする。

【０１３１】好適には、該スケジューリング法は、ストライプグループ中のディスクの各組毎に個別に用いられる。

【０１３２】時間帯を用いてビデオストリームの再生の開始を分配し、これによりディスクアクセスの集中（オーバーロード）を回避する。各ビデオストリームは、特定の時間帯で開始するようにスケジューリングされる（割り当てられる）。該スケジューリング法によれば、Ｍ個の時間帯が存在する（ここで、Ｍはストライプグループ中の記憶装置･･･の数である）。該Ｍ個の時間帯は、$Z_1 \cdots Z_n$と表される。

【０１３３】下記の表１は、１ストライプグループにつき４つの記憶装置･･･を有するシステムにおける好適な時間帯ローテーションを示すものである。

【０１３４】

【表１】

	現在の時間---------->					
	T_1	T_2	T_3	T_4		T_n mod N
ディスク１	Z_1	Z_2	Z_3	Z_4		Z_n mod N
ディスク２	Z_4	Z_1	Z_2	Z_3		$Z_{(n+3)}$ mod N
ディスク３	Z_3	Z_4	Z_1	Z_2		$Z_{(n+2)}$ mod N
ディスク４	Z_2	Z_3	Z_4	Z_1		$Z_{(n+1)}$ mod N

【０１３５】時間は、タイムスロット（T_n）と呼ばれる所定の固定長の時間間隔で測定される。例えば、タイムスロットT_1中に、ディスク１が、時間帯Z_1に割り当てられたビデオストリームのみを開始させ、ディスク２が、時間帯Z_2に割り当てられたビデオストリームのみを開始させる（以下同様）。同様に、タイムスロットT_2中には、ディスク１が、時間帯Z_2に割り当てられたビデオストリームのみを開始させ、ディスク２が、時間帯Z_3に割り当てられたビデオストリームのみを開始させる（以下同様）。従来の方法で行われていたように各ビデオオブジェクトを所定の一定の時間帯（Z_1）に割り当てるのではなく、ビデオストリームが開始することになる記憶装置･･･に関連する最も早い利用可能な時間帯（Z_1）にビデオオブジェクトの再生開始が割り当てられる。最も早い利用可能な時間帯（Z_1）とは、時間帯Z_1に現在割り当てられているビデオストリームに欠陥を決して生じさせることなく再生を操作するのに十分な容量を有する次の時間帯（Z_1）である。

【０１３６】一好適実施例では、Ｍ＝６である。別の実施例では、異なる数の記憶装置･･･が特定のストライプグループに割り当てられている。

【０１３７】図１１は、１ストライプグループ中にＭ個の記憶装置･･･を有するＶＯＤシステムによるスケジューリング法･･･を示すフローチャートである。

【０１３８】該スケジューリング法･･･は、ビデオストリームの再生を開始させるための･･･････メッセージ･･･をストリームスレッド･･･が受信した際に開始する。次いで、ストリームスレッド･･･は、読み出すべき最初のデータブロックを記憶している記憶装置･･･のディスク番号ｎを判定する･ステップ･･･）。次いで、ストリームスレッド･･･は、現在の時間（ｔ）を獲得する･ステップ･･･）。

【０１３９】次いで、記憶スレッド･･･は、現在の時間帯を表すインデックス値（Ｃ）を計算する･ステップ･･

・）。該インデックス値（C）は次の式に従って計算される。

【0140】C＝……t・T・－n・・・・M

ここで、

t＝現在の時間

T＝データブロックを再生するための時間間隔

（即ち、Z＝データブロックサイズ／ストリーム再生データ速度）

n＝ストライプグループ中の記憶装置の番号

M＝ストライプグループ中の記憶装置の総数

・・・・・＝整数値を返すように変数の切り捨てを行って返す関数

該スケジューリング法・・・は、M個の要素を有する時間帯使用列Z・1…M・を用いる。該M個の要素は、それぞれ、最初はゼロにセットされ、対応するM個の時間帯の各々に割り当てられた活動再生ストリームの数を表す。

【0141】ステップ・・・でインデックス値Cを計算した後に、ストリームスレッド・・・が、インデックスIにCをセットする。ストリームスレッド・・・は次いで、時間帯使用列ZのI番目の要素の値を、1つの時間帯に割り当て可能な最大ストリーム数と比較する・ステップ・・・）。1つの時間帯についての最大ストリーム数は、特定の記憶装置・・・に関するアクセス時間によって決まる。該比較ステップ・・・が、時間帯が一杯である（即ち、既に最大ストリーム数を有している）ことを示す結果を返した場合には、該方法は、次の式に従ってインデックス値Iを更新する・ステップ・・・）。

【0142】I＝（I＋1）・・・M

ステップ・・・でインデックス値を更新した後、該方法は比較ステップ・・・に戻る。

【0143】しかしながら、比較ステップ・・・が、時間帯が一杯でないことを示す結果を返した場合には、時間帯使用列Zが更新され・ステップ・・・）、ビデオストリームが時間帯T_Iに割り当てられる・ステップ・・・）。

【0144】ビデオストリームが時間帯T_Iに割り当てられた後、該ビデオストリームは、次の式による時間遅延の後に再生を開始する。

【0145】時間遅延＝（・I＋M＋C・・・・M）＋T

この時間遅延は、所望の（選択された）時間スロットで再生を開始するように導入される。

【0146】ストリームスレッド・・・は、・・・・・・・・・・・・コールを受信した場合、又はストリームの再生を完了した場合には、該再生ストリームに関する使用値Z_Iをデクリメントする。

【0147】RAIDシステム及び方法

VODシステム・・・は、本発明による安価なディスクからなる冗長ディスクアレイ（RAID）システム及び方法を用いる。本発明によれば、記憶モジュール・・・は、複数の記憶装置・・・を使用して複数のビデオオブジェクトを記憶する。本発明によるRAIDシステム及び方法は、ビデオサーバ用途に限定されるものではなく、記憶装置のアレイを使用するあらゆるコンピュータシステム又は構成で有用なものとなる、ということが理解されよう。

【0148】本発明によるRAIDシステム及び方法は、多数のディスクから構成された記憶サブシステム・記憶モジュール・・・）が、データアクセスに関して高いスループットを達成すること、及び、1つ又は2つ以上のディスクが故障した際に失ったデータを動的に再構成する際の性能上の不利益を制限することを可能にする。該システム及び方法は、更に、N個のディスクからなるディスクアレイ中のN／（M＋1）又はそれ未満の個数の記憶装置・・・・ディスク・が故障した場合における動的なデータ再構成を達成することにより連続的な動作を可能にする。ここで、・・・Mは、データオブジェクトがディスクアレイに記憶される際に該データオブジェクトのクリエータにより指定された（又はデフォルト値として割り当てられた）冗長ファクタであり、・・・故障した2つのディスクの距離は、Mよりも大きい。

【0149】該システム及び方法は、N個のディスクへのデータオブジェクトの記憶をインターリーブさせるものである。ここで、Nは、多数の並列ディスク動作を可能にすることにより高性能を得るように、また、M個のデータブロック毎にパリティブロックを作成するために、所望するだけ大きくすることが可能である。なお、Mは、Nよりも小さい整数であり、動的データ再構成時における性能上の不利益を制限するために所望するだけ小さくして（M＝1と選択した場合には、RAIDレベル1ミラーと等価になる）、全ての状況で性能レベルが保証されるようにすることができる。Mが小さいということは、冗長データに関する記憶オーバーヘッドが大きいことを意味する。

【0150】本発明の典型的な用途は、マルチストリームVODシステム・・・であり、この場合、総ディスクスループットは、数十・・・・・・・・・から数百又は数千・・・・・・・・の範囲のものとなる。ビデオサーバ・・・に記憶されている1つのビデオオブジェクトは、数十、数百、又は数千ものユーザにより同時に要求される可能性がある。従って、多数のディスクに対するビデオオブジェクトのストライプ処理が可能であることは必須であり、例えば、・・個のディスクに対してビデオオブジェクトのストライプ処理を行い、・・個のディスク全てが並列動作を行って数百のユーザの要求を満たすようにする。この場合、ビデオオブジェクトに関する冗長ファクタMは、例えば、ディスクが故障した際に失ったデータブロックを再構成するために4個の並列ディスク読み出ししか必要としないように4と選択することが可能である。これは、かかる場合の応答時間を保証するだけでなく、システム全体の作業負荷を殆ど増大させないものとなる。その理由は、それらの4つのディスク読み出しが、失ったデータ

に密接したものであり、及び通常のビデオ再生中にどうしても必要とされるものであり、従ってそれらは（通常のアクセスと比べて）特別なディスク動作ではない、ということにある。この説明のため、アレイ中にN個のディスク（0～Nと番号付けされたもの）が存在するものと仮定する。また、好適には、（ビデオオブジェクト等の）データオブジェクトが作成される際に、ストライプブロックサイズで連続的にデータが分配される（データブロックは0・1・2…と番号付けされている）。

【0151】図12は、本発明に従ってビデオオブジェクトを記憶するRAID方法…を示すフローチャートである。該方法は、最初にセットアップ処理…を行う。セットアップ処理…では、ビデオオブジェクトのクリエータ（例えば、コンピュータプログラム又はユーザ）が該ビデオオブジェクトに関する冗長ファクタMを指定する。Mは、1～N－1の整数である。ここで、Nは、記憶モジュール…中の記憶装置…の個数である。

【0152】次いで、セットアップ処理…中に、該方法は、ビデオオブジェクトの属性として冗長ファクタMを記憶する。該方法は更に、インデックス・I・をゼロに初期化し、及びDRAM…上のパリティバッファを規定し初期化する。

【0153】次いで、該システムは、ビデオオブジェクトに書き込まれるべきデータブロックを取り出す・ステップ…）。各データブロック毎に、該方法は、パリティバッファに対してI番目のデータブロックの排他OR演算を行う・ステップ…）。該方法…は次いで、I番目のデータブロックをJ番目のディスクに書き込む・ステップ…）。ここで、

J＝……I／M…M＋1・＋… … M… … N

である。

【0154】更に、I番目のデータブロックは、J番目のディスク上のビデオオブジェクトのストライプのK番目のブロックとして書き込まれる。ここで、

K＝ ……I／M…M＋1・＋… … M…／N)

である。

【0155】該方法は次いで、現在のデータブロック・I番目のデータブロック・が冗長グループ中の最後のデータブロックであるか否かを判定するためのテストを行う・ステップ…）。該テスト…では下記の判定が行われる。

【0156】…Iが（M－1）以上であるか。

【0157】…（・I＋1・ … M）＝0であるか。

【0158】かかる条件が満たされる場合には、該方法…は、J番目のディスクにパリティバッファを書き込む・ステップ…）。ここで、

J＝・・I＋1・／M・・M＋1・－1・ …・N

である。

【0159】パリティバッファは、J番目のディスク上のデータオブジェクトのストライプのK番目のブロックとして書き込まれる・ステップ…）。ここで、K＝ ……I＋1・／M・・M＋1・－1・／N）である。

【0160】J番目のディスクにパリティバッファを書き込んだ後、該パリティバッファがクリアされる（再び初期化される）・ステップ…）。

【0161】次いで該方法…は、インデックス・I・を1だけインクリメントする・ステップ…）。該方法…は、次いで、ビデオオブジェクトの最後のデータブロックのディスクへの書き込みが完了しているか否かを判定するためのテストを行う・ステップ…）。最後のデータブロックの書き込みが完了していない（即ち、書き込むべきデータブロックが更に存在する）場合には、該方法…は、ステップ…に戻って、ビデオオブジェクトに書き込むべき次のデータブロックを取り出し、該方法…を続行する。また、最後のデータブロックの書き込みが完了している場合には、該方法…は、現在のデータブロック（I番目のデータブロック）が冗長グループ中の最後のデータブロックであるか否かを判定するためのテストへと進む・ステップ…）。該テスト…は、（I … M）を計算することにより行われる。（I … M）がゼロでない場合には、冗長グループはM個未満のデータブロックを有しており、従って、該方法…は、ステップ…へと進んで、全てゼロで満たされた1つのデータブロックをJ番目のディスクに書き込む。ここで、J＝……I／M…M＋1・＋… … M… … Nである。

【0162】ステップ…で、I番目のデータブロックは、J番目のディスク上のデータオブジェクトのストライプのK番目のブロックとして書き込まれる。ここで、

K＝ ……I／M…M＋1・＋… … M…／N)

である。

【0163】次いで、該方法…は、I番目のデータブロックが冗長グループ中の最後のデータブロックであるか否かを判定するためのテストを行う・ステップ…）。次の場合に条件が満たされることになる。

【0164】…Iが（M－1）以上である。

【0165】…（・I＋1・ … M）＝0である。

【0166】該条件が満たされる場合には、該方法…は、J番目のディスクにパリティバッファを書き込む・ステップ…）。ここで、

J＝・・I＋1・／M・・M＋1・－1・ …・N

である。

【0167】更に、パリティバッファは、J番目のディスク上のデータオブジェクトのストライプのK番目のブロックとして書き込まれる・ステップ…）。ここで、

K＝ ……I＋1・／M・・M＋1・－1・／N)

である。

【0168】該方法…は次いでパリティバッファをク

リアし･ステップ･･･）、次いで該データオブジェクトに関するＮ個全てのストライプをクローズする･ステップ･･･）。一方、テスト･･･において条件が満たされなかった場合には、該方法･･･は次いでＩをインクリメントし･ステップ･･･）、次いでステップ･･･に戻って、現在のデータブロック（Ｉ番目のデータブロック）が冗長グループ中の最後のデータブロックであるか否かを判定する。

【０１６９】図１３は、本発明に従ってビデオオブジェクトにアクセスするＲＡＩＤ方法････を示すフローチャートである。該方法････は、Ｊ番目のディスクに記憶されているビデオオブジェクトからＩ番目のデータブロックを読み出すことをストリームスレッド･･･が要求した･ステップ････）際に開始する。該読み出し要求を受けると、該方法････は、ビデオオブジェクトに関する冗長ファクタＭを読み出す･ステップ････）。次いで、該方法････は、故障状態を判定するためのテストを行う･ステップ････）。故障が生じなかったことを該テスト････が示す場合には、該方法････は、適当なディスク（Ｊ番目のディスク）からデータブロックを取り出す。しかしながら、該テスト････が故障が生じたと判定した場合には、該方法････は、データ再構成バッファを全てゼロに初期化する･ステップ････）。次いで、該方法････は、インデックスＰをゼロに初期化する･ステップ････）。Ｐをゼロに初期化することにより、Ｐは、冗長グループ中の最初のデータブロックを指すように初期化される。

【０１７０】次いで、該方法････は、故障したディスクにＰ番目のデータブロックが記憶されていないか判定するためのテストを行う･ステップ････）。故障したディスクにＰ番目のデータブロックが記憶されていると判定された場合には、該方法････は、Ｌ番目の記憶装置上のストライプのＫ番目のデータブロックの読み出しへと処理を進める。ここで、

Ｌ＝･Ｊ＋Ｎ－･Ｉ･･･ Ｍ･＋Ｐ･ ･･･ Ｎ

Ｊ＝･･･････Ｉ／Ｍ･･･Ｍ＋１･＋･Ｉ･･･ Ｍ･･ ･･･ Ｎ

Ｋ＝････････････Ｉ／Ｍ･･･Ｍ＋１･＋･Ｐ ･･･ Ｍ･･／Ｎ）

である。

【０１７１】次いで、該方法は、取り出したデータ及び再構成バッファ中に記憶されているデータの排他ＯＲ演算を行う･ステップ････）。次いで該方法は、ステップ････に進んでインデックスＰをインクリメントする。該インクリメントの後、該方法････は、再構成が完了したか否か（即ち、Ｐ＞Ｍか否か）を判定するためのテストを行う･ステップ････）。再構成が完了している場合には、該方法････は、再構成バッファ中のデータをストリームスレッド･･･に返す。また、再構成が完了していない場合には、該方法････はステップ････に戻る。

【０１７２】上記の説明は、本発明の方法及び実施例の単なる典型例を示すものである。当業者であれば理解されるように、本発明は、その思想及び本質的な特徴から逸脱することなく別の特定形態で実施可能なものである。従って、本発明の開示は、その例示を目的としたものであり、特許請求の範囲に記載の本発明の範囲を限定するものではない。

【図面の簡単な説明】

【図１】本発明によるビデオオンデマンドシステムを示すブロック図である。

【図２】図１のビデオオンデマンドシステムで使用されるプログラムモジュール（処理スレッド）を含む中央制御モジュールを示すブロック図である。

【図３】図２に示す中央制御モジュールで使用される処理スレッドの対話を示す状態図である。

【図４】記憶装置とのアクセスに使用されるデータ構造及びプログラムモジュールを示すフローチャートである。

【図５】図４に示す「要求リスト」の説明図である。

【図６】本発明による図３に示すストリームスレッドの処理状態を示す状態図である。

【図７】各記憶スレッドにより行われるメッセージキュー処理を示すフローチャートである。

【図８】「要求リスト」中のメッセージの記憶スレッドによる処理を示すフローチャートである。

【図９】図１に示す記憶モジュールへの記憶のためのビデオオブジェクトオープン処理を示すフローチャートである。

【図１０】再生のためのビデオオブジェクトオープン処理を示すフローチャートである。

【図１１】図１に示す複数の記憶装置にわたるアクセス負荷の時間的平衡化のためのスケジューリング方法を示すフローチャートである。

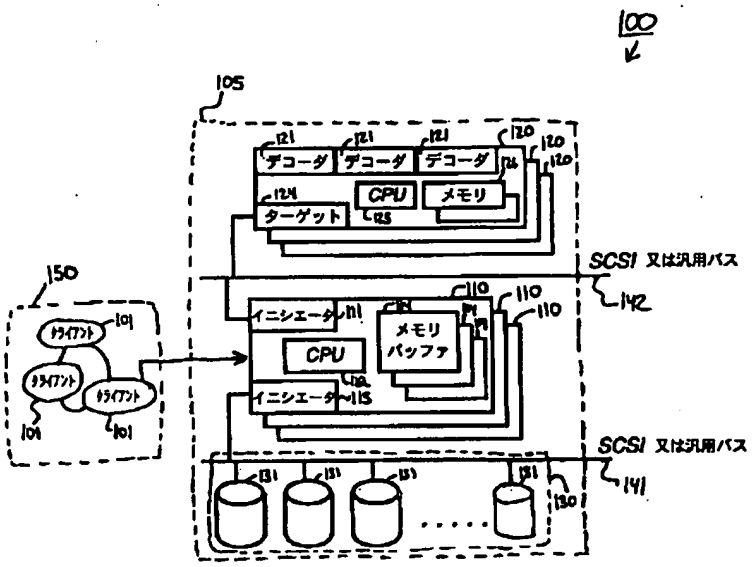
【図１２】Ｍ個のデータブロック毎にパリティコードを生成するように冗長ファクタ（Ｍ）を用いてディスクドライブアレイにビデオオブジェクトを記憶させる方法を示すフローチャートである。

【図１３】図１２に示す方法に従って記憶されたデータブロックを取り出す処理を示すフローチャートである。

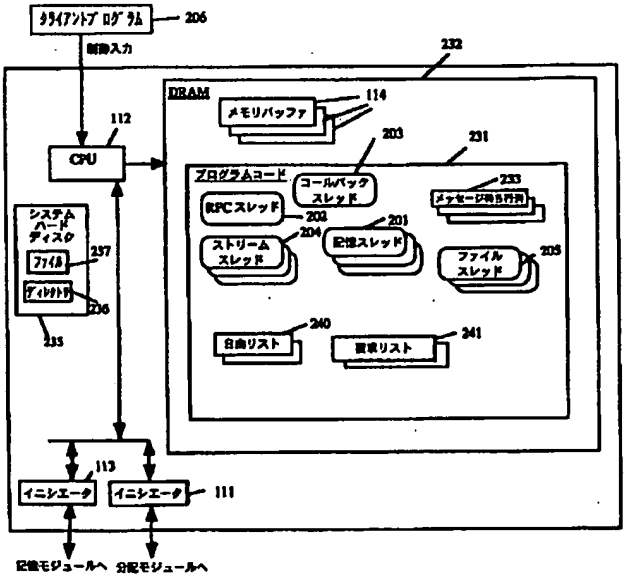
【符号の説明】

･･･　ＶＯＤシステム
･･･　ビデオクライアント
･･･　ビデオサーバ
･･･　中央制御モジュール
･･･　分配モジュール
･･･　記憶モジュール
･･･　制御入力ソース

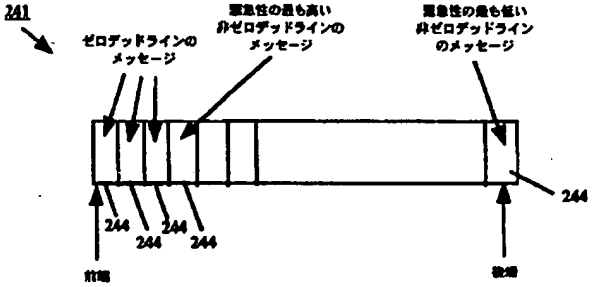
【図1】


100
105
デコーダ
デコーダ
デコーダ
CPU
メモリ
ターゲット
SCSI 又は汎用バス
142
150
イニシエータ
メモリバッファ
CPU
イニシエータ
SCSI 又は汎用バス
141

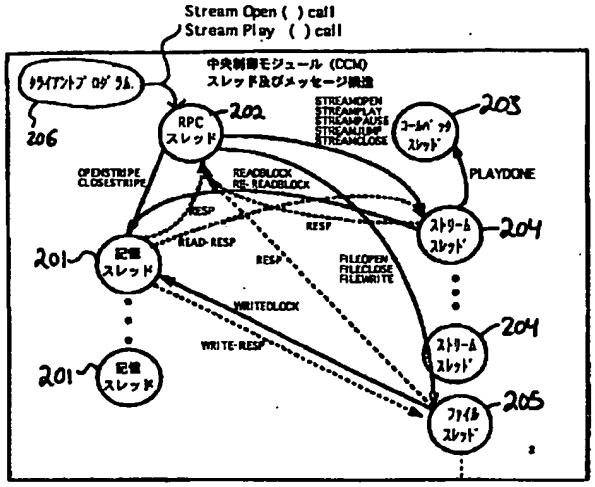
【図2】


DRAM
メモリバッファ
CPU
プログラムコード
コールバックスレッド
RPCスレッド
ストリームスレッド
ファイルスレッド
自由リスト
要求リスト
システムハードディスク
イニシエータ
イニシエータ

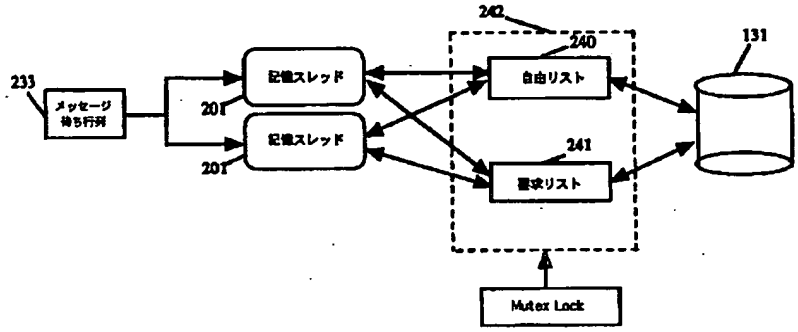
【図5】


241
ゼロデッドラインのメッセージ
244
244
244
244
244

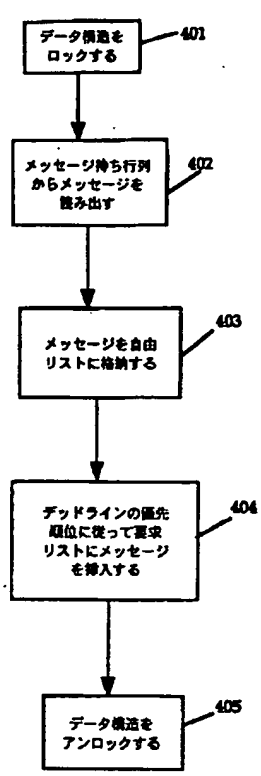
【図3】


Stream Open ( ) call
Stream Play ( ) call
中央制御モジュール (CCM)
スレッド及びメッセージ構造
クライアントプログラム
206
RPC スレッド
202
STREAMOPEN
STREAMPLAY
STREAMPAUSE
STREAMJUMP
STREAMCLOSE
コールバック スレッド
203
PLAYDONE
OPENSTRIPE
CLOSESTRIPE
READBLOCK
RE-READBLOCK
RESP
READ-RESP
ストリーム スレッド
204
201
記憶 スレッド
FILEOPEN
FILECLOSE
FILEWRITE
WRITEBLOCK
WRITE-RESP
ファイル スレッド
205

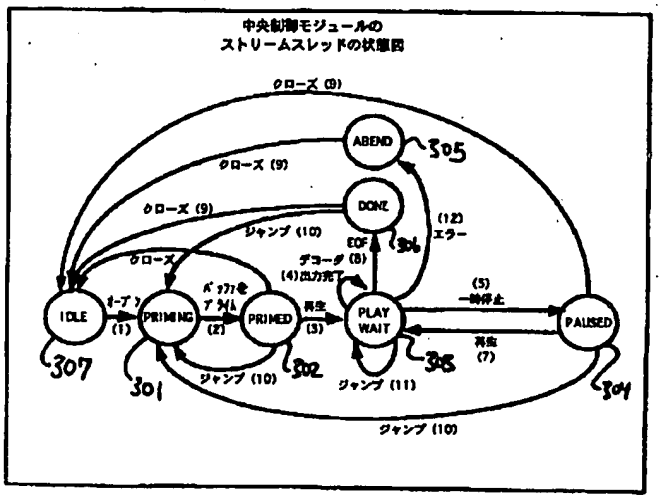
【図4】


233
メッセージ 待ち行列
201
記憶スレッド
記憶スレッド
242
240
自由リスト
241
要求リスト
131
Mutex Lock

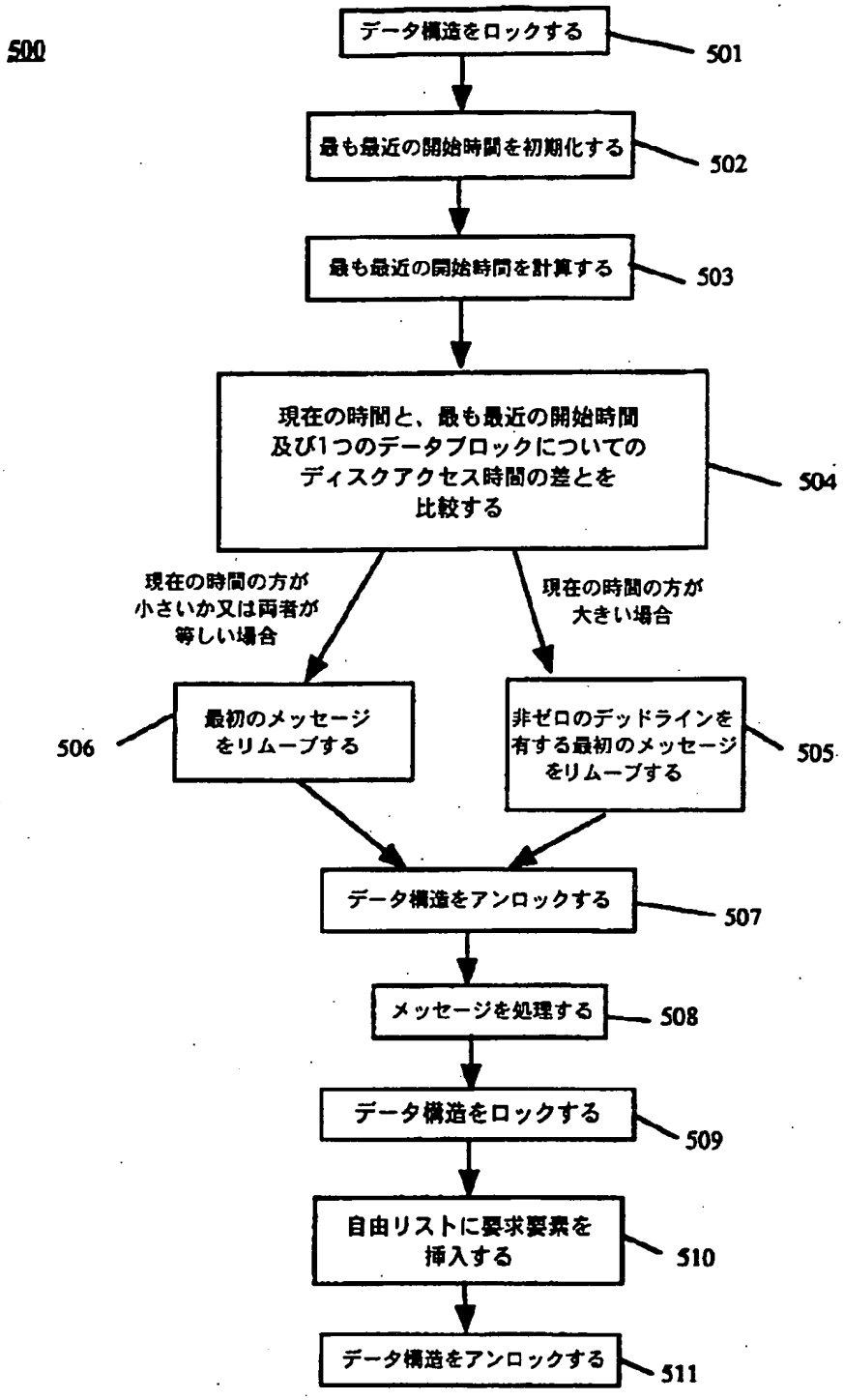
【図7】


400
データ構造を ロックする
401
メッセージ待ち行列 からメッセージを 読み出す
402
メッセージを自由 リストに格納する
403
デッドラインの優先 順位に従って要求 リストにメッセージ を挿入する
404
データ構造を アンロックする
405

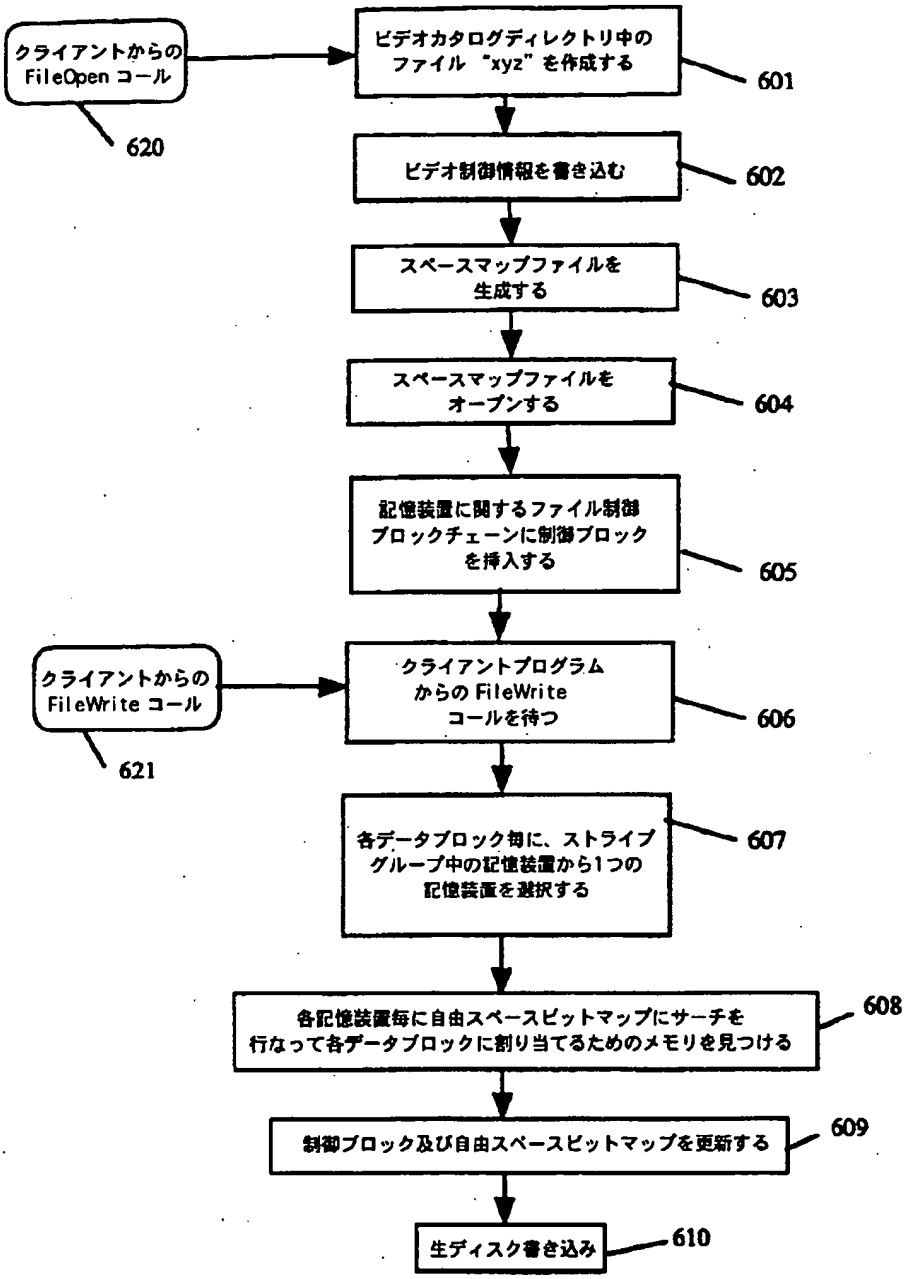
【図6】


中央制御モジュールの
ストリームスレッドの状態図
クローズ (9)
ABEND
305
DONE
(12) エラー
EOF
306
ジャンプ (10)
デコード (8)
(4)出力完了
(5) 一時停止
IDLE
(1)
PRIMING
(2)
PRIMED
(3)
PLAY WAIT
PAUSED
再生 (7)
307
301
302
308
304
ジャンプ (11)

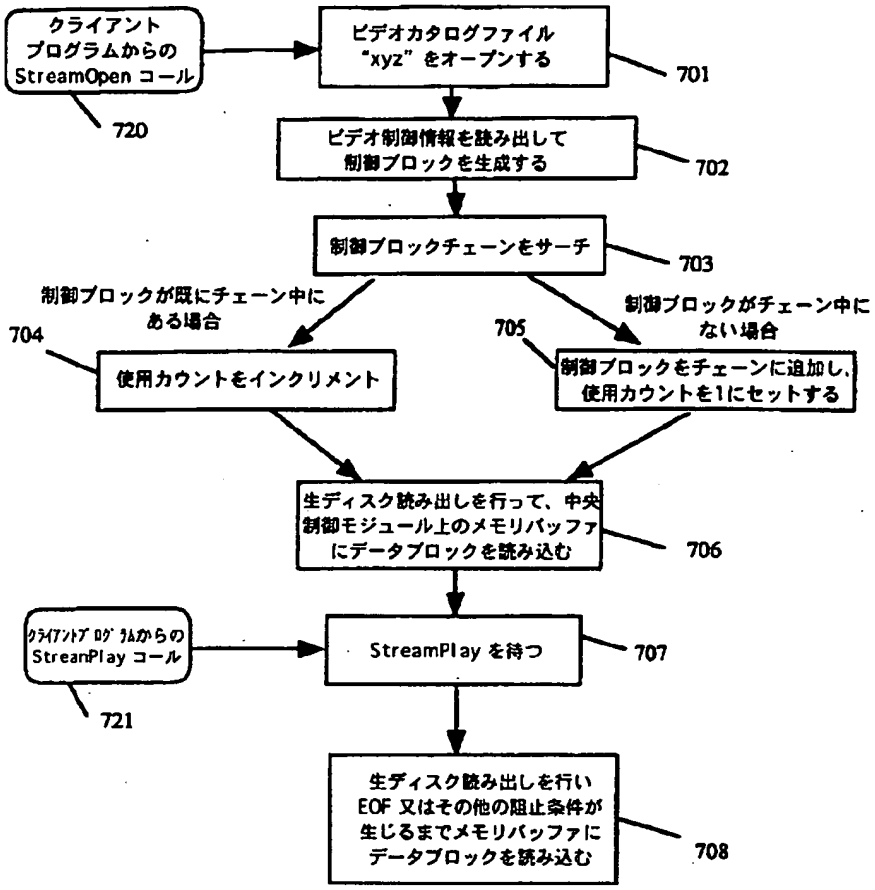
【図８】


500
データ構造をロックする
501
最も最近の開始時間を初期化する
502
最も最近の開始時間を計算する
503
現在の時間と、最も最近の開始時間
及び1つのデータブロックについての
ディスクアクセス時間の差とを
比較する
504
現在の時間の方が
小さいか又は両者が
等しい場合
現在の時間の方が
大きい場合
最初のメッセージ
をリムーブする
506
非ゼロのデッドラインを
有する最初のメッセージ
をリムーブする
505
データ構造をアンロックする
507
メッセージを処理する
508
データ構造をロックする
509
自由リストに要求要素を
挿入する
510
データ構造をアンロックする
511

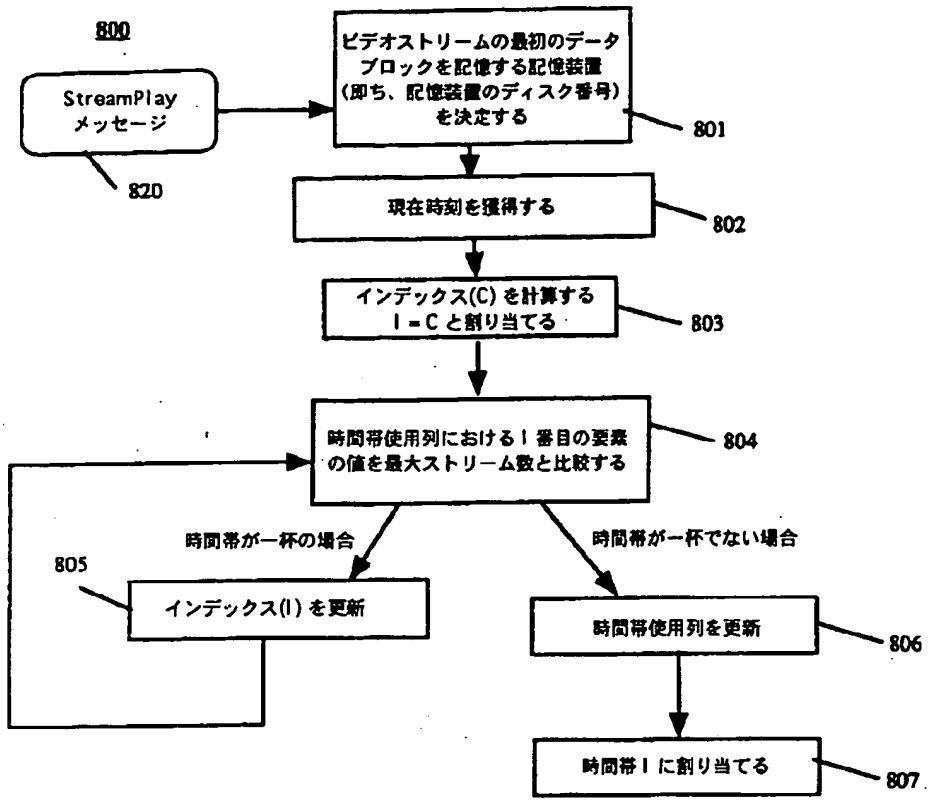
【図9】


クライアントからの FileOpen コール
620
ビデオカタログディレクトリ中のファイル "xyz" を作成する
601
ビデオ制御情報を書き込む
602
スペースマップファイルを生成する
603
スペースマップファイルをオープンする
604
記憶装置に関するファイル制御ブロックチェーンに制御ブロックを挿入する
605
クライアントからの FileWrite コール
621
クライアントプログラムからの FileWrite コールを待つ
606
各データブロック毎に、ストライプグループ中の記憶装置から1つの記憶装置を選択する
607
各記憶装置毎に自由スペースビットマップにサーチを行なって各データブロックに割り当てるためのメモリを見つける
608
制御ブロック及び自由スペースビットマップを更新する
609
生ディスク書き込み
610

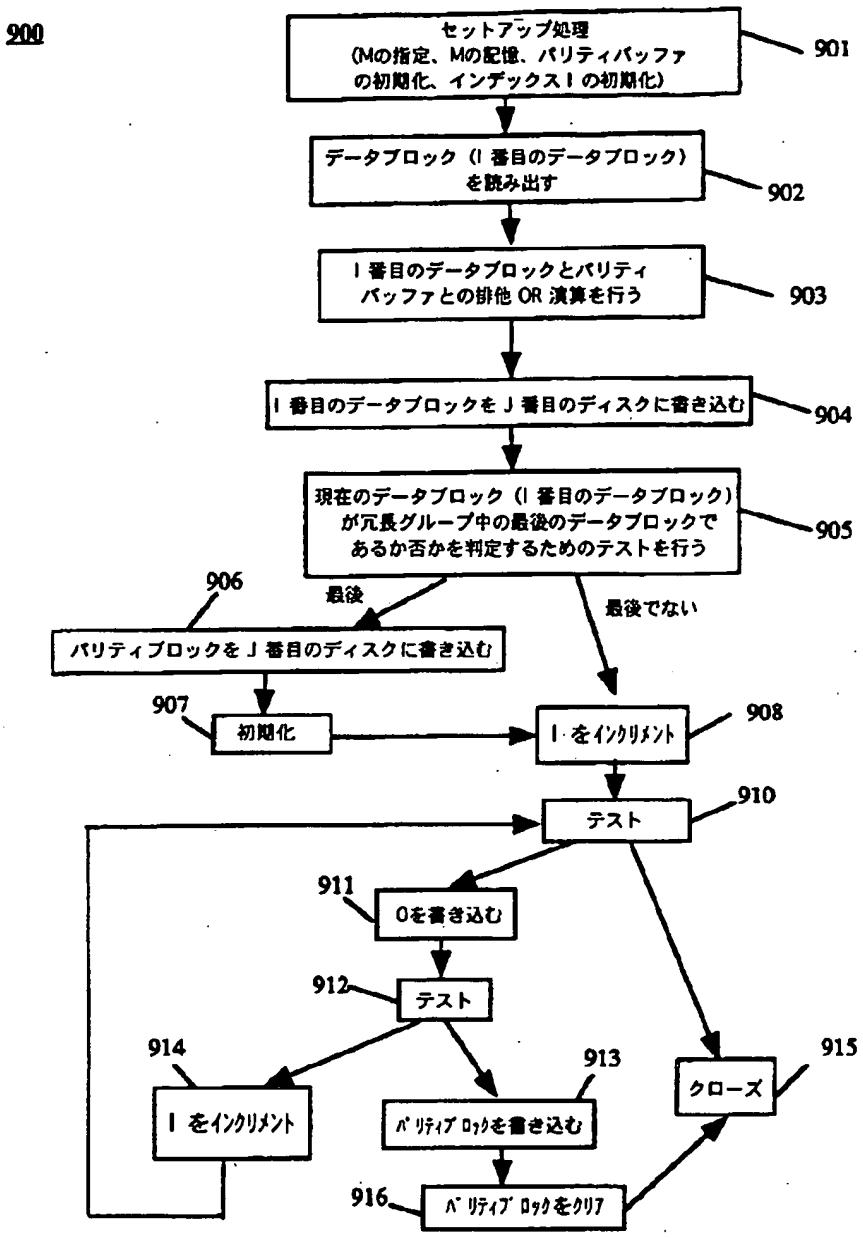
【図１０】


クライアント
プログラムからの
StreamOpen コール
720
ビデオカタログファイル
“xyz” をオープンする
701
ビデオ制御情報を読み出して
制御ブロックを生成する
702
制御ブロックチェーンをサーチ
703
制御ブロックが既にチェーン中に
ある場合
704
使用カウントをインクリメント
制御ブロックがチェーン中に
ない場合
705
制御ブロックをチェーンに追加し、
使用カウントを1にセットする
生ディスク読み出しを行って、中央
制御モジュール上のメモリバッファ
にデータブロックを読み込む
706
ｸﾗｲｱﾝﾄﾌﾟﾛｸﾞﾗﾑからの
StreanPlay コール
721
StreamPlay を待つ
707
生ディスク読み出しを行い
EOF 又はその他の阻止条件が
生じるまでメモリバッファに
データブロックを読み込む
708

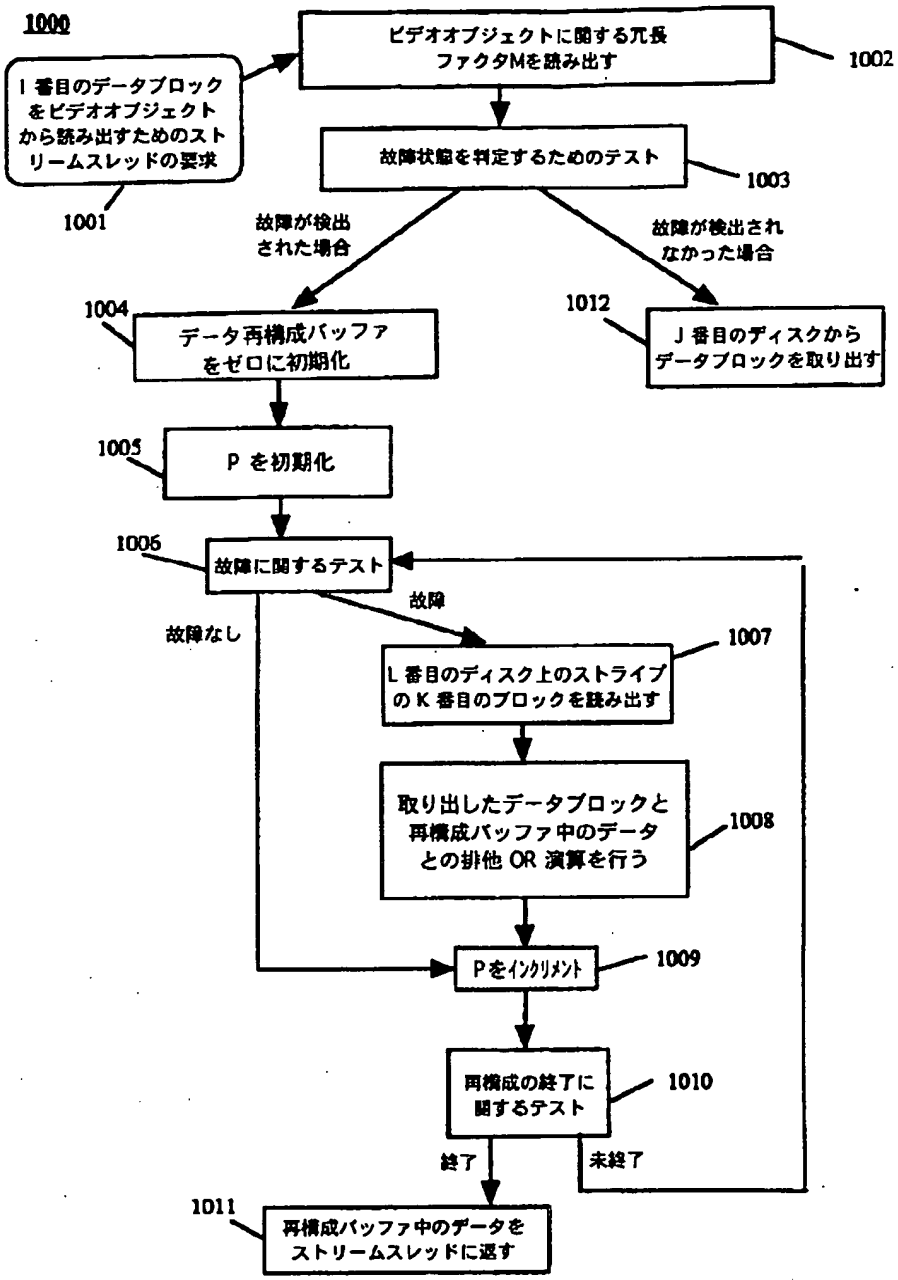
【図11】


800
StreamPlay
メッセージ
820
ビデオストリームの最初のデータ
ブロックを記憶する記憶装置
(即ち、記憶装置のディスク番号)
を決定する
801
現在時刻を獲得する
802
インデックス(C)を計算する
I＝Cと割り当てる
803
時間帯使用列における I 番目の要素
の値を最大ストリーム数と比較する
804
時間帯が一杯の場合
時間帯が一杯でない場合
805
インデックス(I)を更新
時間帯使用列を更新
806
時間帯 I に割り当てる
807

【図１２】


900
セットアップ処理
(Mの指定、Mの記憶、パリティバッファ
の初期化、インデックスIの初期化)
901
データブロック（I番目のデータブロック）
を読み出す
902
I番目のデータブロックとパリティ
バッファとの排他 OR 演算を行う
903
I番目のデータブロックをJ番目のディスクに書き込む
904
現在のデータブロック（I番目のデータブロック）
が冗長グループ中の最後のデータブロックで
あるか否かを判定するためのテストを行う
905
906
最後
最後でない
パリティブロックをJ番目のディスクに書き込む
907
初期化
I をインクリメント
908
テスト
910
911
0を書き込む
912
テスト
914
913
915
I をインクリメント
パリティブロックを書き込む
クローズ
916
パリティブロックをクリア

【図１３】



フロントページの続き

.....6	識別記号	FI	
G06F		G11B	Z
G11B		H04N	A
H04N		.....	
.....		G06F	370D